

Peter Kraft
Regnecentralen A/S
Falkoner alle 1-3.

Pulawy Program System

PULAWY PROGRAM SYSTEM

```

B      17
B      A63, B47, E9
B      I20
      I1=1<22, I2=1<21, I3=1<20, I4=1<19
      I5=1<18, I6=1<17, I7=1<16, I8=1<15
      I9=1<14, I10=1<13, I11=1<12, I12=1<11
      I20=I1~I3~I4~I5~I6~I7~I10~I11

```

```

; THE TAPE CONSISTS OF SEVERAL SEGMENTS

```

```

; EACH ASSOCIATED WITH AN I NAME:

```

```

;      I1          MONITOR
;      I2
;      I3          OPERATOR CONTROL
;      I4          SCAN LOOPS
;      I5          EQUATIONS
;      I6          MACROPROGRAM
;      I7          MACRO EXECUTIVE
;      I8          SIMULATION
;      I9          SIMULATION TABLE
;      I10         SELF CHECK
;      I11         SCAN TABLE
;

```

```

;      FOUR GLOBAL A NAMES TRANSFER THE FOLLOWING
;      BOOLEANS TO EVERY SEGMENT

```

```

;      A55         WHETHER THE SEGMENT MUST BE
;      INCLUDED INTO THE SYSTEM
;      A54         CONTROLS THE JUMP AFTER LOADING
;      THE BINARY SEGMENT LEADING EITHER TO THE
;      OPERATOR CONTROL (BIT (I) = 1) OR THE LOADER
;      TO GO ON READING.
;      A56         ALLOWS SPECIFYING SOME SPECIAL
;      ACTIONS VARYING TO EACH SEGMENT (MAINLY
;      SIMULATION)
;      A58         CONTROLS TEST OUTPUT
;

```

```

;      THE USER CAN SET THE BOOLEANS BY
;      LOGICALLY ADDING THE I-NAME OF THE SEGMENT
;      TO THE RELEVANT A-NAME.

```

```

;      THE INITIAL SITUATIONS:

```

```

A54=I11, A55=I20
A58=0
A56=0

```

```

; DESCRIBE THE NORMAL (OPERATIONAL) VERSION
; OF THE SYSTEM

```

L
; REDEFINE
N

T ; SHIFT TO THE TYPEWRITER, THE OPERATOR NOW
; CAN REDEFINE THE A NAMES IF NECESSARY AND
; RESTART READING WITH P

E
; DEFINITIONS OF A NAMES:

; DEVICE SYMBOLIC ADDRESSES:

A0 = 0 < 6 ; PAPER TAPE READER
A1 = 1 < 6 ; PAPER TAPE PUNCH
A2 = 2 < 6 ; CONTROL TYPE WRITER
A3 = 4 < 6 ; LOG TYPE WRITER 1
A4 = 5 < 6 ; - - - 2
A5 = 3 < 6 ; STRIP PRINTER I + II
A6 = 6 < 6 ; A/D CONVERTER
A7 = 12 < 6 ; DIGITAL PULSE I
A8 = 13 < 6 ; - - II
A9 = 14 < 6 ; - - III

A11 = 8 < 6 ; DIGITAL ALARM I
A12 = 9 < 6 ; - - II
A13 = 10 < 6 ; - - III
A14 = 11 < 6 ; - - IV
A15 = 7 < 6 ; DIGITAL OUTPUT

; IO-COMMANDS:

A20 = 0 ; SENSE
A21 = 1 ; CONTROL
A22 = 2 ; READ
A23 = 3 ; WRITE
A24 = 4 ; NO OPERATION

; ABSOLUTE ADDRESSES OF INTERRUPT VARIABLES:

A25 = 8 ; INTERRUPT NUMBER
A26 = 10 ; INTERRUPTED ADDRESS
A27 = 12 ; INTERRUPT SERVICE ADDRESS
A28 = 14 ; START KEY SERVICE ADDRESS

; MONITOR FUNCTIONS:

A31 = 6 ; IO
A32 = 10 ; EXIT TASK
A33 = 12 ; RESET
A34 = 13 ; START TASK
A37 = 11 ; PASS CONTROL
A38 = 7 ; P OF SEMAPHORES
A39 = 8 ; VR OF SEMAPHORES
A40 = 9 ; V OF SEMAPHORES

; ADDRESS OF MONITOR POINTS:

; A45 ; END OF NAME TAB
; A46 ; FIRST OF NAME TAB
; A47 ; FIRST OF ADDRESS TAB
; A48 ; FIRST OF NEXT EXECUTION TAB
; A49 ; FIRST OF PERIODE TAB

; MONITOR CALL INSTRUCTIONS:

A50 = 11 < 6 ; CALL OF MONITOR FUNCTION
A51 = 22 < 6 ; CALL FOR TEST OUTPUT

; AUXILARY DEFINITIONS:

A60 = 9 ; NO OF TASKS
A61 = 15 ; NO OF DEVICES
A62 = 1 ; BIT FOR CONTROL TWR ALARM LAMP
A63 = 2 ; BIT FOR CONTROL ROOM LAMP

; SYSTEM SEMAPHORES:

```
;
; 1<0 : D, CONTROL TYPEWRITER
; 1<1 : P, IN, INA
; 1<2 : P, CON
; 1<3 : P, VALUE ( SEGMENT 5 )
; 1<4 : P, DEVICE ERROR MESSAGE
; 1<5 : P, GET LOG STRING
; 1<6 : D, LOG TYPEWRITER 1
; 1<7 : D, LOG TYPEWRITER 2
; 1<8 : D, TAPE PUNCH
; 1<9 : D, STRIPPRINTER
; 1<10 : P, DIGITAL VALUE
; 1<11 : T, FLOW INTEGRATION TABLE
; 1<12 : T, PULSE INTEGRATION TABLE
;
; ( D: PERIPHERAL DEVICE RESERVATION
;   P: COMMON PROCEDURE      -
;   T: DATA TABLE          - )
```

; DEFINITIONS OF E NAMES (SYSTEM DATA)

```
E0 = 178      ; NUMBER OF INTEGRATED FLOWS
E1 = 132      ; - - - PULSES
E2 = 270      ; - - BALANCE RESULTS
E3 = 12       ; - - PLANT DATA
E5 = 32       ; - - LINES LOG 1 HEADING
E6 = 14       ; - - LOG 2 -
E7 = 561      ; SCAN NO FIRST DIGITAL ALARM
E8 = 80       ; NUMBER OF INTEGRATED PULSES
              ; BELONGING TO THE FIRST PART
E9 = 134      ; FIRST BALANCE RESULT ON RIGHT SIDE
```

C A55<1, C1

; IF SEGMENT 1 IS NOT TO BE ASSEMBLED, THEN
; THIS BLOCK WILL DEFINE THE B NAMES (SEE
; ALSO SEGMENT 1) AND A45 - A49.

C1 = A28 + 2 ; BEGINNING OF B-TABLE

B39=C1, C1=C1+2, B26=C1, C1=C1+2
B24=C1, C1=C1+2, B25=C1, C1=C1+2
B27=C1, C1=C1+2, B21=C1, C1=C1+2
B19=C1, C1=C1+2

B20=C1, C1=C1+2, B23=C1, C1=C1+2
B44=C1, C1=C1+2, B47=C1, C1=C1+2

B17=C1, C1=C1+2, B18=C1, C1=C1+2
B22=C1, C1=C1+2, B28=C1, C1=C1+4
B29=C1, C1=C1+2, B30=C1, C1=C1+2
B31=C1, C1=C1+2, B32=C1, C1=C1+2
B33=C1, C1=C1+2, B15=C1, C1=C1+2
B34=C1, C1=C1+2, B35=C1, C1=C1+2

B36=C1, C1=C1+2, B37=C1, C1=C1+2
B38=C1, C1=C1+2, B40=C1, C1=C1+2
B41=C1, C1=C1+4, B42=C1, C1=C1+2
B43=C1, C1=C1+2, B45=C1, C1=C1+2
B46=C1, C1=C1+2

; W0, W1, W2, W3, DUMP ACT, X TABLES:

C1=C1+[2×A60+2]×5+[A60+2]^(−2)

; NAME TABLE:

A46=C1, C1=C1+2×A60

A45=C1, C1=C1+2

; ADDR TABLE:

A47=C1, C1=C1+2×A60

; NEXT EXECUTION:

C1=C1-2

A48=C1, C1=C1+2×A60

; PERIOD TABLE:

C1=C1-2

A49=C1

E
L
; END S0
N
T

C [A55>22^1] - 1 ; SEGMENT 1:
S C10, F23

; THE NON-RELOCATABLE FIRST SEGMENT CONTAINS
; THE FOLLOWING PROGRAMS:
; AUTOLOADER (LATER OVERWRITTEN)
; SYSTEM MONITOR
; CONTROL TYPEWRITER PROCEDURES
; LOADER
; AND A TABLE WHICH WILL CONTAIN
; INTER-SEGMENT REFERENCES. (ADDRESS CATALOG)

W K=A25
0 ; INTERRUPT NUMBER
; K=A26
0 ; INTERRUPT ADDRESS
; K=A27
F1 ; INTERRUPT SERVICE ADDRESS
; K=A28
F0 ; START KEY - -

; THE FOLLOWING BLOCK CONTAINS A PROGRAM WHICH
 ; WILL READ IN SEGMENT 1. LOADING OF THE PROGRAM
 ; IS MADE WITH A BOOTSTRAP HEADING WHICH
 ; CAN BE INPUT BY THE AUTOLOAD KEY.
 ; AS THE PROGRAM USES THE AW INSTRUCTION, IF
 ; ANY ERROR OCCURS DURING INPUT, THE MACHINE WILL
 ; STOP, AND THE LOADING MUST BE ATTEMPTED
 ; AGAIN.
 ; THE INPUT TAPE FORMAT IS THE STANDARD
 ; SLANG-BINARY, AND THE CORE STORE ADDRESS
 ; CORRESPONDING K=0 IS ZERO,

H3, J3

S
W

```

AW          2      ; BOOTSTRAP HEADING
AW          4      ;
JL          0      ;
AW          2042   ;
AW          X1+J0   ;
AW          2044   ;
AL W1       X1+2    ;
AW          2046   ;
JL.         -4     ;
JL W1       2042   ;

J1: AW          0      ;
    AW          0      ;
    ML.         H3.    ;
    IC          -1     ;
    AL W0       0      ; SUM:=0;
H0:                                ; LENGTH:
    AW          4      ; LENGTH:= INWORD;
    WA W0       4      ; SUM:= SUM + LENGTH;
    AL W1       X3     ; SAVE:= K;
    AW          6      ; K:= INWORD;
    WA W0       6      ; SUM:= SUM + K;
    SH W2       -1     ; IF LENGTH < 0 THEN
    JL.         H1.    ; GO TO END;
H2:                                ; NEW WORD:
    SN W2       0      ; IF LENGTH = 0 THEN
    JL.         H0.    ; GO TO LENGTH;
    AW          X3     ; STORE[K]:= INWORD;
    WA W0       X3     ; SUM:= SUM + STORE[K];
    AL W3       X3+2   ; K:= K + 2;
    AL W2       X2-2   ; LENGTH:= LENGTH - 2;
    JL.         H2.    ; GO TO NEW WORD;
H1:                                ; END:
    PC          0      ; UNPROTECT REGISTERS;
    PC          2      ;
    PC          4      ;

```

```
PC          6      ;
SN  W0      0      ;   IF SUM = 0 THEN
JL          X1-2    ;   GO TO STORE[SAVE - 2];
JL.         H1.    ;   GO TO END;
H3:  -1-1<22      ;   MASK OVERFLOW INTERRUPT
                                     ;
J2:  JL.          J1. ;
                                     ;
J0=2042-J2+J1-4    ;
```

E

```

; ADDRESS CATALOG:
; EVERY QUANTITY WHICH IS REFERENCED IN
; MORE THAN ONE SEGMENT HAS GOT AN ENTRY
; IN THE FOLLOWING TABLE. THE TABLE CAN BE ADDRESSED
; ABSOLUTELY. THE ENTRIES CONTAIN THE QUANTITY
; ITSELF, OR AN ABSOLUTE ADDRESS WHERE THE
; QUANTITY CAN BE FOUND, AND ARE INITIALIZED
; WHEN THE SEGMENT CONTAINING THE QUANTITY IS
; LOADED
;   FORMAT OF THE TABLE COMMENTS:
;   <TYPE>, <INITIALIZING SEGMENT NO> <DESCRIPTION>
;   <TYPE> CAN BE ADDRESS, VARIABLE OR PROCEDURE

```

C8:

W

; PROTECTED ROUTINES:

B0:

```

B39: C0      ; P, 1  LOADER
B26: C1      ; P, 1  OUTA
B24: C2      ; P, 1  CON
B25: C3      ; P, 1  INA
B27: C4      ; P, 1  IN
B21: C5      ; P, 1  ERROR MESSAGE
B19: C6      ; P, 1  CLOCK TO HOUR MIN
B1:

```

; GENERAL PURPOSE PROCEDURES:

```

B20: 0      ; P, 5  VALUE
B23: 0      ; P, 5  NEXT ENTRY
B44: 0      ; P, 7  MACRO EXECUTIVE
B47: 0      ; P, 4  BALANCE RESET PHASE

```

; OTHER ITEMS:

```

B17: 0      ; A, 1  LAST LOADED WORD
B18: 0      ; A, 8  SIMULATION TABLE BASE
B22: 0      ; A, 6  MACROPROGRAM
H B28: 0, 0  ; A, 11 H1, H2
      0      ;      H3
W B29: 0      ; A, 7  BALANCE RESULT TABLE
B30: 0      ; A, 11 SCAN TABLE
B31: 0      ; A, 11 SCAN TABLE PART 2 REL
B32: 0      ; A, 11 DIGITAL ALARM SCAN TABLE
B33: 0      ; A, 11 DIGITAL PULSE SCAN TABLE PART 1
B15: 0      ; A, 11 - - - - - PART 2
B34: 0      ; A, 7  FLOW INTEGRATION TABLE
B35: 0      ; A, 7  PULSE INTEGRATION TABLE

B36: -1<10  ; V, 3  ACTIVE GROUPS
B37: 1<9~1<7~1<3
      ; V, 3
B38: 0      ; V, 1  CLOCK

```

B40: 0 ; V, 4 NUMBER OF INTEGRATION STEPS
B41: <DDMM>
 <YYYY> ; V, 3 DATE
B42: 1 ; V, 3 SCAN NO TO BE TREND LOGGED
B43: 0 ; V, 4, 1 LIGHTENED LAMPS

B45: 0 ; V, 7 LAST RESET AND
B46: 0 ; V, 7 PERIODE FOR BALANCE CALCULATION
C9:

; IN B36 THE I-TH BIT DESCRIBES THE
; STATE OF GROUP I:
; BIT = 0 GROUP IS PASSIVE
; BIT = 1 GROUP IS ACTIVE
; IN B37 THE BITS DESCRIBE THE SO-CALLED
; RUN MODES:
; BIT 23 = 1 BALANCE: PRINT ON LOG TWR 1
; - 22 = 1 - : - - LOG TWR 2
; - 21 = 1 - : - - PUNCH
; - 20 = 1 LOG 1 : - - LOG TWR 1
; - 19 = 1 - : - - LOG TWR 2
; - 18 = 1 - : - - PUNCH
; - 17 = 1 LOG 2 : - - LOG TWR 1
; - 16 = 1 - : - - LOG TWR 2
; - 15 = 1 - : - - PUNCH
;
; - 14 = 1 BALANCE: RESET
; - 13 = 1 - : CALCULATION
;
;


```
B      C10, D23, G30      ; BEGIN
                                ; MONITOR:
                                ; COMMENT:

;      THE MONITOR CONSIST OF A SET OF PROGRAMS WITH
;      THE FOLLOWING ENTRIES:
; F0    START KEY
; F1    INTERRUPT SERVICE
; F2    PROGRAM ERROR
; F3
; F4    PRINT EXIT
; F5
; F6    IO
; F7    P OF SEMAPHORES
; F8    VR OF SEMAPHORES
; F9    V OF SEMAPHORES
; F10   EXIT TASK
; F11   START KEY INTERRUPT
; F12   CLOCK INTERRUPT
; F13   TIMER INTERRUPT
; F14   BUSY RETURN
; F15   PASS CONTROL
; F16   RESET
; F17   START TASK
; F18   NEXT IN TURN 1
; F19   NEXT IN TURN 2
; F20   RESTART 1
; F21   RESTART 2
; F22   SET MODE
```

; TABLES USED IN MONITOR:

; COMMENT THE DUMP TABLE, W0-TAB, W1-TAB, W2-TAB, W3-TAB,
; DUMP ACTION, AND X-TAB, ARE USED FOR SAVING OF REGISTERS
; AND RETURN ADDRESS, WHEN A TASK PROGRAM IS INTERRUPTED
; AND CONTROL IS TRANSFERRED TO THE MONITOR. DUMP ACTION
; CONTAINS EITHER THE RETURN ADDRESS, OR IF THE CORRESPONDING
; TASK PROGRAM IS PASSIVE, ZERO.

; THERE ARE AN ENTRY IN THE TABLES FOR EACH
; TASK PROGRAM, PLUS ONE EXTRA ENTRY, WHICH IS USED
; IF AN EXTERNAL INTERRUPT MAKES REQUEST IN THE INTERVAL
; WHERE SWITCHING IS MADE FROM LAST TASK TO TASK 0,
; INTERRUPT IS ENABLED.

; THE EXTRA ENTRIES ARE ALSO USED DURING TEST OUTPUT,
; WHEN THE MONITOR IS CALLED FROM THE STANDARD ROUTINES.

; THE NAME-TAB AND ADDR-TAB CONTAINS AN ENTRY
; FOR EACH TASK, WHERE THE NAME, GIVEN AS FOUR
; CHARACTERS, AND THE INITIALIZE ADDRESS OF THE TASK
; PROGRAMS ARE STORED. THE NAME TAB CONTAIN FURTHER
; AN ENTRY USED FOR LOOK UP OF NAMES.

; NEXT EXECUTION AND PERIODE ARE TABLES WHICH
; CONTAIN AN ENTRY FOR EACH TASK BUT TASK 0. THE
; TABLES CONTAIN THE TIME FOR NEXT EXECUTION OF A TASK
; PROGRAM AND THE PERIODE FOR REPETITIONS.

; STATUS TAB IS USED FOR INPUT OUTPUT. IT
; CONTAINS AN ENTRY FOR EACH DEVICE. AN ENTRY CONTAINS
; EITHER AN ADDRESS WHERE THE RESULT OF AN IO-
; OPERATION SHALL BE STORED OR A ZERO IF THE OPERATION
; ON THE DEVICE HAS BEEN TERMINATED BY A
; SENSE COMMAND.

W GO=K;

; ARRAY W0-TAB [0:NO OF TASKS × 2];
C0: 0,0,0,0,0,0,0,0,0,0,0,0,0,0
K = GO + 2 × A60 + 2, GO = K

; ARRAY W1-TAB [0:NO OF TASKS × 2];
C1: 0,0,0,0,0,0,0,0,0,0,0,0,0,0
K = GO + 2 × A60 + 2, GO = K

; ARRAY W2-TAB [0:NO OF TASKS × 2];
C2: 0,0,0,0,0,0,0,0,0,0,0,0,0,0
K = GO + 2 × A60 + 2, GO = K

; ARRAY W3-TAB [0:NO OF TASKS × 2];
C3: 0,0,0,0,0,0,0,0,0,0,0,0,0,0
K = GO + 2 × A60 + 2, GO = K

; ARRAY DUMP ACTION [0:NO OF TASKS × 2];
C4: 0,0,0,0,0,0,0,0,0,0,0,0,0,0
K = GO + 2 × A60 + 2, GO = K

; ARRAY X-TAB [0:NO OF TASKS];
H C5: 0,0,0,0,0,0,0,0,0,0,0,0,0,0
K = GO + [A60 + 2] ^ [-2], GO = K

; ARRAY NAME-TAB [0:NO OF TASKS × 2]
W A46: ; FIRST IN NAME TAB
C6: -1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1
K = GO + 2 × A60
A45: 0 ; LAST IN NAME TAB
GO = K

; ARRAY ADDR TAB [0:NO OF TASKS × 2 - 2];
A47: ; FIRST IN ADDR TAB
C7: F2,F2,F2,F2,F2,F2,F2,F2,F2,F2,F2,F2,F2,F2
K = GO + 2 × A60, GO = K

; ARRAY NEXT EXECUTION [2: NO OF TASKS × 2 - 2];
K = K - 2, GO = K
A48: ; FIRST IN NEXT EXECUTION;
C8: K = K + 2 ; TASK 0 HAS NO ENTRY
1<22,1<22,1<22,1<22,1<22,1<22,1<22,1<22,1<22,1<22,1<22
K = GO + 2 × A60 , GO = K

; ARRAY PERIOD [2:NO OF TASKS × 2 - 2];
K = K - 2, GO = K
A49:

; FIRST IN PERIOD

C9: K = K + 2 ; TASK 0 HAS NO ENTRY

1,1,1,1,1,1,1,1,1,1,1

K = GO + 2 * A60, GO = K

; ARRAY STATUS TAB [0:NO OF DEVICES - 1];

H C10: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

K = GO + [A61+1] ^ [-2]

```
; VARIABLES USED IN MONITOR:
W D0: 0 ; INTEGER MODE;
; COMMENT TASK NO IN BITS 0:22
; MODE = SINGLE: BIT 23 = 0
; MODE = MULTI: BIT 23 = 1;

; D1 ; BOOLEAN T;
; COMMENT HALF WORD IN TIMER INTERRUPT;

D2: 0 ; BOOLEAN ARRAY SEMAPHORES[1:23];

D3: 0 ; INTEGER TEMP;

; CONSTANTS USED IN MONITOR:
D4: -1
H D5: A50 , G18 ; MONITOR CALL LOWER BOUND
D6: A50 , G19 ; MONITOR CALL UPPER BOUND
W D7: JL W3 (B0) ; SUBROUTINE CALL LOWER BOUND
D8: JL W3 (B1) ; SUBROUTINE CALL UPPER BOUND

D9: 1<22 ; INFINITE
H D11: 0 , -64 ; MASK IO DEVICE NO
W D12:
D10: 63 ; MASK CHAR
D13: 86400 ; DAY IN SECONDS
D14: 10 ; START KEY INTERRUPT NUMBER

; TEXTS USED IN MONITOR:
; STRING ERROR HEAD =
W D22: <DDDD:<58>> ; ADDR STRING CAT
D23: <AAAA:<58><10>> ; NAME STRING;

H D15: <PROGRAM ERROR,<10>> ;
D17: <EXIT,<10>>
D19: <START KEY,<10>>

; STRING TEST OUTPUT =
D20: <DDDDDDDD> ; TEST STRING NUM CAT
D21: <AA<63><10>> ; TEST STRING AN
```

; INTERRUPT SERVICE:

; COMMENT THE MONITOR IS ALWAYS CALLED BY INTERRUPT, IN-
; TERRUPT SERVICE EXAMINES THE REASON FOR INTERRUPT AND TRANS-
; FERS CONTROL TO A RELEVANT MONITOR ACTION, FOLLOWING IS
; PERFORMED:
;

; 1. W-REGISTERS, EXCEPTION REGISTER, AND INTERRUPTED ADDRESS
; ARE DUMPED TO THE DUMPTABLE CORRESPONDING TO
; INTERRUPTED TASK.
;

; 2. FOR INTERRUPT NUMBER 3 AND 4 MONITOR PROGRAM
; FOR HANDLING CLOCK INTERRUPT OR TIMER INTERRUPT
; IS CALLED.
;

; FOR INTERRUPT NUMBER 1 AND 2 ERROR ACTIONS ARE
; TAKEN.
;

; INTERRUPT NUMBER 5 IS SIMULATED FOR THE START KEY,
;

; FOR INTERRUPT NUMBER 0 THE INSTRUCTION WHICH CAU-
; SED THE INTERRUPT IS EXAMINED.
;

; FOLLOWING DISTINCTION IS MADE:
;

; 3. CALL OF MONITOR FUNCTION: A RELEVANT MONITOR PROGRAM
; IS CALLED.
;

; 4. CALL OF A PROTECTED SUBROUTINE FROM AN UNPROTECTED
; INSTRUCTION: TRANSFER FOR TO THE SUBROUTINE IS MADE.
;

; 5. CALL OF MONITOR FOR PRINTING TEST OUTPUT: IF THE
; MODE IS SINGLE TEST OUTPUT IS PRINTED WITH THE
; FOLLOWING FORMAT:
;

; <VALUE OF W0><AA><CR>

; WHERE <AA> IS TWO ALPHANUMIC CHARACTERS GIVEN AS
; CALL PARAMETER, EACH PRINTING IS TERMINATED BY A
; CAR RETURN.
;

; 6. INSTRUCTIONS WHICH CAUSE INTERRUPT OF OTHER REASON
; THAN THE ABOVE ARE REGARDED AS ERRORS, AND THE
; ERROR PROGRAM IS CALLED;
;

```

B      H3, I1                      ; BEGIN
W F0:  RX, W2                      D14, ; START KEY:
      RS  W2                      A25  ; INTERRUPT NUMBER:=10;
      RX, W2                      D14, ;
F1:    AM, (D0,)                  ; DUMP OF REGISTERS:
      RS, W2                      C2,  ; W2 TAB [MODE]:= W2;
      RL, W2                      D0,  ; W2:= MODE;
      RS, W1                      X2+C1, ; W1 TAB [MODE]:= W1;
      RS, W3                      X2+C3, ; W3 TAB [MODE]:= W3;
      RS, W0                      X2+C0, ; W0 TAB [MODE]:= W0;
      LD  W3                      -1,  ;
      XS, X2+C5,                  ; X TAB [MODE]:= EXCEPTION REG;
      LD  W3                      1,   ;
      RL  W3                      A26  ; W3:= INTERRUPTED ADDRESS;
      RS, W3                      X2+C4, ; ACTION TAB [MODE]:= W3;
      RL  W1                      A25  ; TEST INTERRUPT NUMBER:
      LS  W1                      -1,  ;
      SE  W1                      0,   ; W1 = INTERRUPT NUMBER:2;
      JL, H1,                     ; IF W1 = 0 THEN GOTO SWITCH;
                                   ; TEST MONITOR FUNCTION CALL;
                                   ; COMMENT FORMAT OF MONITOR CALL:
                                   ; A50, <FUNCTION NUMBER>;
      RL  W1                      X3-2, ; W1:= WORD(MONITOR CALL);
      SL, W1                      (D5,) ; IF W1 < MONITOR CALL LOWER BOUND
      SL, W1                      (D6,) ; ~ W1 > MONITOR CALL UPPER BOUND
      JL, H2,                     ; THEN GOTO TEST SUBROUTINE TRANSFER;
      BZ  W1                      3,   ; W1:= FUNCTION NUMBER;

H1:    BL, W1                      X1+3, ; SWITCH:
G1:    JL, X1,                     ; GOTO CASE W1 OF
H      G2, ; = G20 -G1, ; OVERFLOW ERROR,
      G3, ; = G21 -G1, ; FLOATING ERROR,
      G6, ; = F12 -G1, ; CLOCK INTERRUPT,
      G7, ; = F13 -G1, ; TIMER INTERRUPT,
G16:   G4, ; = F11 -G1, ; START KEY INTERRUPT; COMMENT LAST INTERRUPT
      G8, ; = F6 -G1, ; 10,
      G9, ; = F7 -G1, ; P OF SEMAPHORES,
      G10, ; = F8 -G1, ; VR OF SEMAPHORES,
      G11, ; = F9 -G1, ; V OF SEMAPHORES,
      G12, ; = F10 -G1, ; EXIT TASK,
      G13, ; = F15 -G1, ; PASS CONTROL,
      G14, ; = F16 -G1, ; RESET,
G17:   G15, ; = F17 -G1, ; START TASK; COMMENT LAST FUNCTION;

                                   ; TEST SUBROUTINE TRANSFER;
                                   ; COMMENT FORMAT OF SUBROUTINE CALL:
                                   ; JL W3 (<ADDRESS CATALOG ENTRY>);

```

```

W H2: SL. W1      (D7.) ; IF W1 < SUBROUTINE CALL LOWER BOUND
      SL. W1      (D8.) ; ~ W1 > SUBROUTINE CALL UPPER BOUND
      JL.         H3. ; THEN GOTO TEST FOR TEST OUTPUT;
      HS. W1      I1. ;
      RS. W3      X2+C3. ;
H      AL. W3      (, I1: 0) ; W3:=ADDRESS CATALOG[ADDRESS CATALOG ENTRY]
W      JL.         F21. ; GOTO RESTART 2;

; TEST FOR TEST OUTPUT;
; COMMENT FORMAT OF TEST OUTPUT CALL:
; A51, <AN CHARACTERS>;
H3: HS. W1      D21. ; TEST STRING AN:= AN CHARACTERS;
    BL. W1      2 ;
    SE. W1      A51 ; IF BITS(0,11)OF:(W1) = A51
    JL.         F2. ; THEN GOTO PROGRAM ERROR;
    SZ. W2      1 ; IF MODE = MULTI
    JL.         F21. ; THEN GOTO RESTART 2;
    AL. W2      A60*2 ; W2:= MODE;
    RX. W2      D0. ; MODE:= TASK TOP SINGLE;
    JE. W3      (B24) ; ENABLE INTERRUPTS;
H      D20. , 8 ; CONV AN (W0, TEST STRING NUM);
W      AL. W0      D20. ; OUT AN (W0, TEST STRING);
      JE. W3      (B26) ;
      RS. W2      D0. ; MODE:= W2;
      JD.         F20. ; GO DISABLE RESTART 1;

E ; END INTERRUPT SERVICE;

```


; PROGRAMMING ERRORS:

; COMMENT OVERFLOW INTERRUPT, AND FLOATING INTERRUPT ARE
; REGARDED AS PROGRAM ERRORS.; EXIT FROM A TASK PROGRAM IN SINGLE MODE
; WILL CALL FOR EXIT PRINTING.; AFTER PRINTING CONTROL IS TRANSFERED TO TASK 0,
; THE OPERATOR CONTROL PROGRAM, IN SINGLE MODE,

; FORMAT OF PRINTING:

; <ADDRESS><NAME><TEXT><CR>

; WHERE: ADDRESS IS THE ABSOLUTE ADDRESS + 2 OF THE INSTRUCTION
; WHICH CAUSED THE INTERRUPT,

; NAME IS THE NAME OF THE TASK PROGRAM;

```

B      H1                      ; BEGIN
W F2:                      ; PROGRAM ERROR:
G20:                      ; OVERFLOW ERROR:
G21: AM                      G23 ; FLOATING ERROR:
                                ; <PROGRAM ERROR.>
F11: AM                      G24 ; START KEY INTERRUPT:
                                ; <START KEY.>
F4:  AL  W0                  G25 ; PRINT EXIT:
                                ; <EXIT.>

      HS. W0                  H0. ; TEST ADDR:= W0;
      AL W2                    0  ; W2:= MODE;
      RX. W2                   D0. ; MODE:= TASK 0 + SINGLE;
      RL. W0                   X2+C4. ;
      JL W3                    (B24) ; ADDR STRING:=CON(
H      D22. , 4                ; ACTION TAB[W2]);
W      RL. W0                   X2+C6. ; NAME STRING:=
      RS. W0                    D23. ; NAME TAB[W2];
      AL. W0                    D22. ;
      JL W3                    (B26) ; OUT AN (ERROR HEAD);
H G22: AL. W0 , H0: 0          ; INTEGER TEXT ADDR;
W      JL W3                    (B26) ; OUT AN (TEST ADDR);
      JE.                      (C7,) ; GO ENABLE TASK 0;

E                                ; END PROGRAMMING ERROR;

```

; IO:

; COMMENT

; CALL: W0 = PARAMETER,

; <BUSY RETURN>: A50, A31

; <STATUS ADDR>., <DEVICE NO + COMMAND>

; <NORMAL RETURN>:

; FIVE DIFERENT COMMANDS CAN BE EXECUTED FOR EACH DEVICE:
; SENSE, CONTROL, READ, WRITE, AND NO OPERATION.; THE PARAMETER STATUS ADDR. CONTAINS THE ADDRESS RELATIVE
; TO THE PARAMETER WHERE THE RESULT OF THE COMMAND EXECUTION
; SHALL BE STORED. THIS ADDRESS IS TRANSFORMED TO AN ABSO-
; LUTE WORD ADDRESS AND SAVED IN THE STATUS TABLE.; IN GENERAL A COMMAND IS EXECUTED IN THE FOLLOWING
; STEPS:; 1. IF THE STATUS TABLE ENTRY CORRESPONDING TO THE DEVICE
; IN QUESTION CONTAINS AN ADDRESS THE DEVICE IS SENSED,
; IF THE DEVICE IS BUSY, BUSY RETURN IS MADE, OTHER-
; WISE THE RESULT OF SENSING IS STORED IN THE WORD
; GIVEN BY THE ADDRESS IN THE STATUS TABLE.

; 2. THE COMMAND IS THEN EXECUTED.

; 3. FINALLY THE PARAMETER STATUS ADDRESS IS SAVED
; IN THE STATUS TABLE, AND NORMAL RETURN IS MADE.

; THE STEPS EXECUTED FOR THE DIFFERENT COMMANDS ARE:

; SENSE: STEP 1, AND THEN A ZERO IS STORED IN THE STATUS
; TABLE, NORMAL RETURN.

; CONTROL, READ, AND WRITE: ALL STEPS ARE EXECUTED.

; NO OPERATION: ONLY STEP 1 AND STEP 3 IS EXECUTED,

; THE PARAMETER IN W0 IS ONLY RELEVANT FOR THE
; COMMANDS CONTROL AND WRITE,

; DEVICE ERRORS:

; DISCONNECTED DEVICES WILL SET THE STATUS BIT DATA LOST,
; ERROR DUE TO WRONG DEVICE NUMBER WILL CAUSE A
; PROGRAM ERROR.; WRONG COMMAND OR WRONG PARAMETER IN W0
; WILL SET STATUS BITS AS RESULT OF COMMAND
; EXECUTION.; NO CHECK IS MADE WHETHER A STATUS ADDRESS
; FROM A NON PROTECTED AREA REFERS TO A PROTECTED
; AREA;

B H3, I2 ; BEGIN

W F6: BZ W1 X3+1 ;

```

SL W1      14<6+5      ; IF DEVICE NO>14 THEN
JL.        F2.        ; GO TO PROGRAM ERROR;
HS. W1      12.        ; I2:= DEVICE NO + COMMAND;
LA. W1      D11.       ;
HS. W1      I1.        ; I1:= DEVICE NO;
LS W1      -6         ; W1:= DEVICE NO;
BZ. W2      X1+C10.    ; W2:= STATUS TAB[W1];
SN W2      0          ; IF W2 = 0 THEN
JL.        H1.        ; GOTO EXAMINE COMMAND;
LS W2      1          ; TEMP:= W2 * 2;
RS. W2      D3.       ; COMMENT TEMP = OLD STATUS ADDR;
;
; SENSE WAITING STATUS:
H   IO W2    , I1: 0   ; INTEGER I1; SENSE(W2, I1);
W   SX      1         ; IF BUSY THEN
JL.        F14.       ; GOTO BUSY RETURN;
RS. W2      (D3.)     ; WORD [TEMP]:= W2;
;
H1: BZ W2    X3+1      ; EXAMINE COMMAND:
LA. W2      D10.      ; W2:= COMMAND;
SN W2      0          ; IF COMMAND = SENSE
JL.        H3.        ; THEN GOTO STORE ADDR;
SL W2      A24        ; IF COMMAND > NO OPERATION
JL.        H2.        ; THEN GOTO GET ADDR;
H   IO W0    , I2: 0   ; INTEGER I2; IO(W0, I2);
W   SX      1         ; IF BUSY THEN
JL.        F14.       ; GO TO BUSY RETURN;
;
H2: BL W2    X3        ; GET ADDR:
WA W2      6          ; W2:= STATUS ADDR;
LS W2      -1         ; W2:= W2 : 2;
H3: HS. W2    X1+C10.  ; STORE ADDR:
; STATUS TAB[W1]:= W2;
RL. W2      D0.       ; W2:= MODE;
AL W3      X3+2       ; W3:= W3 + 2; COMMENT RETURN ADDR;
JL.        F21.       ; GOTO RESTART 2;
E                                     ; END;

```

; P OF SEMAPHORES:

; COMMENT

; CALL: <BUSY RETURN>: A50, A38
; <SEMAPHORE MASK>

; <NORMAL RETURN>:

; IN SINGLE MODE THE MONITOR FUNCTION IS DUMMY.
; THE SEMAPHORES, STORED IN D2, ARE COMPARED WITH THE
; SEMAPHORE MASK, IF NONE OF THE BITS ARE SET IN THE MASK, WHICH
; ARE SET IN THE SEMAPHORES THEN THE MASK IS LOGICAL ADDED
; TO THE SEMAPHORES, AND NORMAL RETURN IS MADE.
; OTHERWISE BUSY RETURN IS MADE.

```

B      H3                                ; BEGIN
                                ; COMMENT BLOCK FOR P, VR, AND V;
W F7:  RL. W0                        D2.  ; W0:= SEMAPHORES;
      SO  W2                        1    ; IF MODE = MULTI THEN
      JL. H1.                      ; BEGIN
      SZ  W0      (X3)              ; IF -, SEMAPHORE MASK ^ W0 = ZERO
      JL. F14.                     ; THEN GOTO BUSY RETURN;
      LO  W0      X3                ; W0:= W0 v SEMAPHORE MASK,
H1:    AL  W3      X3+2              ; W3:= W3 + 2; COMMENT SETTING RETURN;
      JL. H3.                      ; GOTO STORE;

```

; VR OF SEMAPHORES;

; COMMENT

; CALL: A50, A39
; <SEMAPHORE MASK>
; <RETURN ADDRESS>; THE SEMAPHORES CORRESPONDING TO THE BITS IN THE MASK
; ARE RESET, RETURN IS MADE TO THE ADDRESS GIVEN AS
; PARAMETER TO THE CALL;F8: RL W0 X3 ; W0:= SEMAPHORE MASK;
RL W3 X3+2 ; W3:= WORD[W3+2]; COMMENT INDIRECT RETURN;
JL. H2. ; GOTO SET V;

; V OF SEMAPRORES:

; COMMENT

; CALL: A50, A40
; <SEMAPHORE MASK>
; <RETURN>:; SAME AS VR, ONLY RETURN IS MADE TO THE WORD FOLLOWING
; SEMAPHORE MASK;F9: RL W0 X3 ; W0:= SEMAPHORE MASK;
AL W3 X3+2 ; W3:= W3 + 2; COMMENT NORMAL RETURN;
H2: LX. W0 D4. ; SET V: W0 := -, W0;
LA. W0 D2. ; W0:= W0 ^ SEMAPHORES;
H3: RS. W0 D2. ; STORE: SEMPHORES:= W0;
JL. F21. ; GOTO RESTART 2;

E ; END BLOCK FOR P , VR, AND V;

```
; EXIT TASK:

; COMMENT
; CALL:      A50, A32
;           DUMP ACTION FOR THE TASK IS SET TO <PASSIVE>.
;           IN SINGLE MODE JUMP IS MADE TO PRINT EXIT
;           WHERE AN EXIT MESSAGE IS GIVEN ON THE CONTROL TYPEWRITER,
;           AND CONTROL IS TRANSFERED TO TASK 0.
;           IN MULTI MODE JUMP IS MADE TO NEXT IN
;           TURN 1, WHERE THE BOOLEAN T IS MADE FALSE AND
;           SWITCHING IS PERFORMED TO THE NEXT TASK PROGRAM;

W F10: AL  W3      0      ; DUMP ACT [W2]
      RS. W3      X2+C4.  ;      := <PASSIVE>;
      SO  W2      1      ; IF MODE = SINGLE
      JL.         F4.    ; THEN GOTO PRINT EXIT
      JL.         F18.   ; GOTO NEXT IN TURN 1
```

; CLOCK INTERRUPT:

; COMMENT A CLOCK INTERRUPT ARRIVES EACH SECOND. THE ADDRESS
; CATALOG VARIABLE, CLOCK, IS COUNTED UP BY ONE, THUS
; CONTAINING THE CURRENT VALUE OF TIME IN SECONDS.; IN MULTI MODE IT IS EXAMINED WHETHER THERE ARE
; ANY OF THE TASK PROGRAM WHICH SHALL BE ACTIVATED,
; A PROGRAM IS NOT INITIATED IF IT IS NOT <PASSIVE>, OR
; ITS VARIABLE, NEXT EXECUTION, IS EITHER <INFINITE> OR
; GREATER THAN THE CLOCK. IF A PROGRAM SHALL RUN ITS
; DUMP ACTION IS SET TO <ENTER>, AND ITS VARIABLE
; NEXT EXECUTION IS COUNTED UP BY THE PERIODE FOR
; THE PROGRAM UNTIL NEXT EXECUTION IS GREATER
; THAN THE CURRENT VALUE OF THE CLOCK.; IN MULTI MODE IT IS FURTHER EXAMINED
; WHETHER THE VALUE OF CLOCK EXCEEDS A DAY. IF IT IS SO THE
; VALUES OF ALL NEXT EXECUTIONS WHICH ARE NOT <INFINITE>
; ARE DECREASED BY A DAY IN SECONDS. THE CLOCK IS
; DECREASED BY A DAY, AND THE ALPHANUMERIC VALUE
; OF THE CURRENT DAY, KEPT IN THE LOAD CATALOG IS
; INCREASED BY ONE;

B H6 ; BEGIN

```

W F12: RX W3      B38      ; W3 CROSS CLOCK;
      AL W3      X3+1      ; W3:= W3 + 1;
      SO W2      1        ; IF MODE = SINGLE
      JL.        H6.      ; THEN GOTO STORE CLOCK;
      AL W2      0        ; W2:= 0; COMMENT TASK 0;
H1:    AL W2      X2+2      ; STEP: W2:= W2 + 2;
      SN W2      A60 x 2    ; IF W2 = NO OF TASK
      JL.        H4.      ; THEN GOTO FIN LOOP;
      RL W1      X2+C8.    ; W1:= NEXT EXECUTION[W2];
      SN W1      (D9.)     ; IF W1 = <INFINITE>
      JL.        H1.      ; THEN GOTO STEP;
      RL W0      X2+C4.    ; W0:= DUMP ACT [W2];
      SN W0      0        ; IF W0 = <PASSIVE>
      SL W1      X3+1      ;   W1 > W3
      JL.        H3.      ; THEN GOTO TEST DAY;
      RL W0      X2+C7.    ; DUMP ACT[W2]:=
      RS W0      X2+C4.    ; ADDR TAB[W2];
      AL W0      1        ;
      SL W0      (X2+C9.)  ; IF PERIODE[W2]<1 THEN
      RS W0      X2+C9.    ; PERIODE[W2]:=1;
H2:    WA W1      X2+C9.    ; FOR W1:= W1 + PERIODE [W2]
      SH W1      X3        ; WHILE W1 < W3 DO;
      JL.        H2.      ;
H3:    SL W3      (D13.)    ; TEST DAY: IF W3 > DAY IN SECONDS
      WS W1      D13.      ; THEN W1:= W1 - DAY IN SECONDS;

```

```
RS. W1      X2+C8.      ; NEXT EXECUTION[W2] := W1;
JL.         H1.        ; GOTO STEP;
H4: RL. W2      D0.      ; FIN LOOP: W2:= MODE;
SL. W3      (D13.)     ; IF W3 < DAY IN SECONDS
JL.         H5.        ;
JL.         H6.        ; THEN GOTO STORE CLOCK;
H5: BZ W0      B41      ;
LA. W0      D12.      ;
AL W1       1         ; INCREASE DATE;
SL W0       16        ;
AL W1       1-16     ;
SN W0       9         ;
AL W1      16-9+64    ;
BA W1      B41      ;
HS W1      B41      ;
WS. W3      D13.     ; W3:=W3-DAY IN SECONDS;
H6: RX W3      B38     ; STORE CLOCK: W3 CROSS CLOCK;
JL.         F21.     ; GOTO RESTART 2;
E           ; END;
```


; TIMER INTERRUPT:

; COMMENT TIMER INTERRUPT ARRIVES EACH 20 MS. IN SINGLE MODE
; THE INTERRUPT IS DUMMY.

; A BOOLEAN, T, TELLS WHETHER CONTROL SHALL BE TRANSFERED
; TO THE NEXT PROGRAM IN THE MULTIPLEX CHAIN.

F13: SO W2 1 ; IF MODE = MULTI
JL. F21. ; THEN GOTO RESTART 2;
AL W0 0 ;
H SN W0 , D1: 0 ; BOOLEAN T; IF T
W JL. F19. ; THEN GOTO NEXT IN TURN 2;
HS. W0 D1. ; T:= TRUE;
JL. F21. ; GOTO RESTART 2;

; BUSY RETURN:

; COMMENT THE PROGRAM IS JUMPED TO WHEN A BUSY SITUATION
; HAS BEEN DETECTED, EITHER IN THE IO-PROGRAM OR IN
; P OF SEMAPHORES.

; IF MODE IS SINGLE CURRENT TASK IS RESTARTED,
; AND OTHERWISE THE NEXT PROGRAM IN THE MULTIPLEXING CHAIN
; IS STARTED.

F14: RL. W2 D0. ; W2:= MODE;
RL. W3 X2+C4. ; W3:= DUMP ACTION [MODE];
AL W3 X3-2 ; W3:= W3 - 2; COMMENT RETURN
; ADDRESS IS SET TO REPEAT THE CALL;

; PASS CONTROL:

; COMMENT

; CALL: A50, A37

; IF MODE IS SINGLE THE CALL IS DUMMY. OTHERWISE
; SWITCHING IS MADE TO THE NEXT TASK

F15: SO W2 1 ; IF MODE = SINGLE
JL. F21. ; THEN GOTO RESTART 2;
RS. W3 X2+C4. ; DUMP ACTION [MODE]:= W3;
JL. F18. ; GO TO NEXT IN TURN 1;

```

; RESET
; COMMENT
; CALL:      A50, A33
;           THE PROGRAM SET DUMP ACTIONS FOR ALL TASKS
; TO <PASSIVE>, THE SEMAPHORES ARE RESET, AND MODE
; IS SET TO SINGLE, TASK 0, CONTROL IS TRANSFERED TO
; TASK 0;

B      H1                                ; BEGIN
W F16: AL  W0                            0 ;
      AL  W2      A60x2                  ;   FOR W2:= LAST TASK STEP -2
H1:    AL  W2      X2=2                    ;   UNTIL TASK 0 DO
      RS. W0      X2+C4.                  ;   DUMP ACTION [W2]:= <PASSIVE>;
      SE  W2      0                      ;
      JL.          H1.                    ;
      RS. W2      D2.                    ;   SEMAPHORES:= 0;
F22:   RS. W2      D0.                    ;   MODE:= TASK 0 SINGLE;
      JE.          (C7,)                 ;   GO ENABLE ADDR TAB [TASK 0];
E                                ; END;

```

```

; START TASK;
; COMMENT
; CALL:      W0 = TASK + SINGLE OR MULTI
;           A50, A34
;           IT IS TESTED WHETHER W0 CONTAINS A PROPER
; TASK NUMBER. IF THE DUMP ACTION FOR THE TASK IS
; <PASSIVE> THE DUMP ACTION IS CHANGED TO <ENTER>.
; THE DUMP ACTION FOR THE TASK IS EXECUTED;

```

```

F17: SL  W0      0                      ; BEGIN IF W0 < TASK 0
      SL  W0      A60x2                  ;   v W0 > MAX TASK NO.
      JL.          F2.                    ;   THEN GOTO PROGRAM ERROR;
      RL  W2      0                      ;   W2:= W0;
      RL  W3      X2+C4.                  ;   W3:= DUMP ACT [W2];
      SN  W3      0                      ;   IF W3 = <PASSIVE>
      RL  W3      X2+C7.                  ;   THEN W3:= ADDR TAB [W2];
      RS. W2      D0.                    ;   MODE:= W2;
      JL.          F21.                  ;   GOTO RESTART 2;

```

; EXIT MONITOR:

; COMMENT THE PROGRAM CONTAINS THE CODE FOR MULTIPLEX
; SWITCHING AMONG THE TASKS.

; WHEN A SWITCHING HAS BEEN DONE, THE DUMP
; ACTION FOR THE NEXT TASK IS EXAMINED. IF THE TASK
; IS PASSIVE, (DUMP ACTION = 0) SWITCHING IS MADE TO
; THE NEXT TASK. OTHERWISE THE TASK IS ACTIVE, THE
; REGISTERS FOR THE TASK ARE RESTORED, AND CONTROL
; IS GIVEN TO THE TASK PROGRAM AT THE ADDRESS
; KEPT IN THE DUMP ACTION.

; AS THE MONITOR WORKS IN DISABLED MODE
; IT IS NECESSARY, WHEN ALL TASKS ARE PASSIVE TO
; ENABLE INTERRUPT FOR SOME PERIODE IN A SWITCHING
; CYCLE, OTHERWISE CLOCK AND KEY INTERRUPT WOULD
; NEVER COME THROUGH, AND NO TASK PROGRAM WOULD
; EVER BE STARTED. WHEN SWITCHING IS MADE FROM
; THE LAST TASK TO TASK 0, INTERRUPT IS ENABLED.
; SPECIAL ENTRIES IN THE DUMP TABLES EXIST FOR
; THIS PURPOSE.

; THERE ARE TWO SET OF ENTRIES TO THE EXIT
; MONITOR PROGRAM. NEXT IN TURN ENTRIES ARE JUMPED
; TO IN MULTI MODE.

; NEXT IN TURN 1: THE BOOLEAN T IS SET FALSE AND

; NEXT IN TURN 2: SWITCHING IS MADE TO THE NEXT
; TASK.

; RESTART ENTRIES ARE USED WHEN THE CURRENT INTERRUPT-
; TED PROGRAM SHALL BE RESTARTED.

; RESTART 1: THE MODE IS SAVED AND THE
; RETURN ADDRESS IS TAKEN FROM THE
; DUMP ACTION TABLE TO W3, AND

; RESTART 2: THE RETURN ADDRESS IS IN W3. THE
; REGISTERS ARE RESTORED, AND CONTROL
; IS TRANSFERED TO THE TASK PROGRAM AT
; THE RETURN ADDRESS;
; REG: W2 = TASK AND MODE;

B H2 ; BEGIN

W H1: AL W2 X2=A60X2-2 ; TOP OF CYCLE: W2:= W2 - NO OF TASK * 2;
JD. 2 ; DISABLE INTERRUPT;

H2: ; PASSIVE;

F18: AL W1 1 ; NEXT IN TURN 1;
HS. W1 D1. ; T:= FALSE;

F19: AL W2 X2+2 ; NEXT IN TURN 2: W2:= W2 + 2;
F20: RS. W2 D0. ; RESTART 1: MODE:= W2;

```

SL W2      A60x2      ; IF W2 = NO OF TASK
JE.         H1.       ; THEN GO ENABLE TOP OF CYCLE;
RL W3      X2+C4.     ; W3:= DUMP ACTION[W2];
SN W3      0          ; IF W3 = 0
JL.         H2.       ; THEN GOTO PASSIVE;

F21: RS W3      D3.    ; RESTART 2: REG W2 = TASK AND MODE,
LD W3      -1        ; W3 = RETURN ADDR;
XL.        X2+C5.     ; TEMP:= W3;
LD W3      1          ; EXEPTION REG:= X TAB[W2 : 2];
RL W0      X2+C0.     ; W0:= W0 TAB [W2];
RL W1      X2+C1.     ; W1:= W1 TAB [W2];
RL W3      X2+C3.     ; W3:= W3 TAB [W2];
RL W2      X2+C2.     ; W2:= W2 TAB [W2];
JE.        (D3,)     ; GO ENABLE WORD[TEMP];

```

```

E                                     ; END EXIT MONITOR;

```

```

; PROGRAM POINTS AND DEFINITIONS;

```

```

; G0 USED FOR DUMP TABLE DEFINITIONS
; G1 INTERRUPT SERVICE SWITCH;
G2 = G20 - G1 ; OVERFLOW ERROR ACTION
G3 = G21 - G1 ; FLOATING ERROR ACTION
G4 = F11 - G1 ; START KEY INTERRUPT
G6 = F12 - G1 ; CLOCK INTERRUPT ACTION
G7 = F13 - G1 ; TIMER INTERRUPT ACTION
G8 = F6 - G1 ; IO ACTION
G9 = F7 - G1 ; P OF SEMAPHORES ACTION
G10 = F8 - G1 ; VR OF SEMAPHORES ACTION
G11 = F9 - G1 ; V OF SEMAPHORES ACTION
G12 = F10 - G1 ; EXIT TASK ACTION
G13 = F15 - G1 ; PASS CONTROL ACTION
G14 = F16 - G1 ; RESET ACTION
G15 = F17 - G1 ; START TASK ACTION
; G16 SWITCH LAST OF EXTERNAL INTERRUPT;
; G17 SWITCH LAST OF MONITOR FUNCTION CALL;
G18 = G16 - G1 ; MONITOR FUNCTION LOWER BOUND;
G19 = G17 - G1 ; MONITOR FUNCTION UPPER BOUND;
; G20 OVERFLOW ERROR
; G21 FLOATING ERROR
; G22 PROGRAM ERROR: INTEGER TEXT ADDRESS

G25 = D17 - G22 ; STRING EXIT,
G24 = D19 - D17 ; START KEY,
G23 = D15 - D19 ; PROGRAM ERROR;

```

```

E
L
; END S1 FIRST PART
N
T

```



```

; THE LOADER SERVES FOR LOADING ANY BINARY
; TAPE. THE ADDRESS CORRESPONDING K = 0 ON THE
; TAPE IS GIVEN AS PARAMETER, IN W0. W1 TELLS
; WHETHER THE LOADED PROGRAM MUST BE
; PROTECTED OR NOT (W1 < 0 = PROTECTION). IF W2
; NOT EQUAL TO ZERO, THE LOADER AFTER MEMORIZING
; THE OTHER PARAMETERS, TRANSFERS THE CONTROL TO
; THE OPERATORS PROGRAM.
; READER ERROR CAUSES A JUMP TO OPERATOR
; CONTROL, AFTER THE USUAL ERROR MESSAGE.
; THE TAPE READ IN MUST HAVE A PROPER TAIL,
; THAT IS:
; 1. THE ENTRY POINT TO THE TAIL IS THE LAST
; WORD ON THE TAPE
; 2. THE TAIL INITIALIZES EVERY GLOBAL LOCATION,
; REFERRING TO ANY PART OF THE TAPE.
; IF THE TAPE IS TO BE INCLUDED INTO THE
; SYSTEM, THEN THE LOGICAL END OF ITS TAIL
; MUST BE:
; AL W0 <FIRST FREE WORD>.
; AL W1 0
; AL W2 A54><N>^1
; JL W3 (B39)
; WHERE <N> = 23 - SEGMENT NUMBER
; CALL: JL W3 (B39)

```

```

B      H10, I1
W
C0:    RS W0      B17 ; LOADER:
      SE W2      0   ; LAST := FIRST FREE;
      JL.        H4. ; IF DO NOT READ THEN
      AL W3      , PC X3 ; GO TO OPERATORS PROGRAM;
W      SH W1      -1 ;
H      AL W3      , PS X3 ; PROTECTION:= W1 < 0
W      HS W3      H7. ;
      AL W2      0   ; READ:
      RS W2      H3. ; COUNT:=
H5:    SN W2      0   ; SUM:= 0;
      AL W2      8   ; NEXT WORD:
      AL W1      0   ; IF COUNT = 0 THEN
H6:    AL W1      0   ; LENGTH:= TRUE;
      AL W2      X2+4 ; NEW:
      AL W2      X2+4 ; WORD:= 0;
      AL W2      X2+4 ; FOR I:= 4 STEP -1 UNTIL 1 DO
H8:    LS W1      6   ; BEGIN
      A50      , A31 ; WORD:= WORD SHIFT 6;
H      I1.      , A22+A0 ;

```

b2 w0
h1: al w1

```

      A50      , A31      ;
      I1.      , A20+A0   ;
W     RL. W0    I1.      ;
      SH W0     63      ; IF ERROR THEN
      JL.      H2.      ; BEGIN
H1:   LD W1     48      ; BAD: JL W3 (B21) ; ERRORMES
H4:   JL       (A47)    ; GO TO OPERATORS PROGRAM
      ;               END;
W H2: WA W1     0      ; WORD:= WORD + INCHAR
      AL W2     X2-1    ;
      SZ W2     7      ;
      JL.      H8.      ; END;
      RL. W0     H3.      ;
      WA W0     2      ; SUM:= SUM + WORD;
      RS. W0     H3.      ;
      SZ W2     63      ; IF -, (LENGTH ~ KDEF ~ CHECK) THEN
      JL.      H9.      ; BEGIN
H7:   RS W1     X3      ; STORE[K]:= WORD;
      PC       X3      ;
      AL W3     X3+2    ; K:= K + 2;
      AL W2     X2-64   ; COUNT:= COUNT - 1;
      JL.      H5.      ; GO TO NEXT WORD
      ;               END;
H9:   SO W2     8      ; IF LENGTH THEN
      JL.      H10.     ; BEGIN
      AL W2     X1      ; LENGTH:= CHECK:= KDEF:= FALSE;
      LS W2     5      ; COUNT:= WORD : 2;
      ;               IF WORD < 0 THEN
      ;               CHECK:= TRUE ELSE
      AL W2     X2+16   ; KDEF:= TRUE;
      JL.      H6.      ; GO TO NEW
      ;               END;
H10:  RX W1     6      ; EXCHANGE(WORD, K)
      WA W3     B17     ; K:= K + LAST;
      AL W2     X2-16   ; IF KDEF THEN
      SO W2     32      ; BEGIN KDEF:= FALSE;
      JL.      H5.      ; GO TO NEXT WORD
      ;               END;
      RS W1     B17     ; LAST:= WORD;
      SN W0     0      ; IF SUM = 0 THEN
      JL       X1-2    ; GOTO STORE[WORD - 2];
      JL.      H1.      ; GO TO BAD;
H3:   0      ; INTEGER SUM,
I1:   0      ;
E

```

; THE FOLLOWING PROCEDURES ORGANIZE THE
; HANDLING OF THE CONTROL TYPEWRITER.
; TO MAKE IT POSSIBLE FOR THIS UNIT SERVING
; INPUT AND OUTPUT PURPOSES AS WELL, THE
; FOLLOWING RULES HAVE BEEN ADOPTED:
; 1. IN MULTI MODE THE TYPEWRITER INPUT
; IS AVAILABLE IN ONE AND ONLY ONE TASK PROGRAM.
; 2. IF AN INPUT ROUTINE HAD BEEN CALLED,
; AND THERE HAS BEEN NO CHARACTER TYPED
; FOR TEN SECONDS, THE INPUT PROGRAM RELEASES
; THE TYPEWRITER SEMAPHOR, AND GIVES A TURN
; IN THE MULTIPLEXER SWITCH, IF THERE IS ANY
; OUTPUT REQUEST, IT CAN BE ACCEPTED IN
; THIS TIME.

B J6

W ; PROCEDURE SET ALARM LAMP;
; CALL JL. W1 J0.

J0: AL W0 A62 ; BEGIN
C7: LO W0 B43 ;
H A50 , A31 ; LIGHT ON(LIGHTENED LAMPS-TW ALARM)
J1. , A15+A23 ;
W JL X1 ; END;

```

;      PROCEDURE OUTA OUTPUTS AN ALPHANUMERIC
; TEXT FROM A GIVEN ADDRESS UP TO BUT NOT
; INCLUDING AN END CHARACTER 10. THE OUTPUT
; IS ALWAYS PRECEDED BY A LC AND A RED RIBBON
; CHARACTER, AND FOLLOWED BY A BLACK RIBBON CHA-
; RACTER. THE PROGRAM LITS ON ALARM LAMP
; IF PARITY ERROR OCCURS (OR THE THE TEXT
; CONTAINS CHARACTERS WHICH ARE NOT IN ALPHABET)

```

B H4, I2

```

W      ; PROCEDURE OUTA(TEXT ADDRESS);
; REG W0 TEXT ADDRESS / DESTROYED
;      W1 SAVED
;      W2 SAVED
;      W3 RETURN / DESTROYED
; CALL JL W3 (B26)

```

H C1: A50 , A38 ; BEGIN

W 1<0 ; P(0);

RS. W3 I0. ;

RS. W1 I1. ;

RS. W2 I2. ;

AL W2 (0) ;

AL W0 29 ;

JL. W1 H3. ; WRITECHAR(RED RIBBON);

JL. W1 H2. ; WRITECHAR(LC);

AL W3 0 ; LEFT:= FALSE

H1: AC W3 X3+6 ; NEW;

; LEFT:= -, LEFT;

BZ W0 X2 ; CHAR:= STORE[TEXT ADDRESS];

LS W0 X3+18 ; IF LEFT THEN CHAR:= CHAR SHIFT -6;

LS W0 -18 ; CHAR:= CHAR ^ 63;

SN W3 0 ; IF -, LEFT THEN

AL W2 X2+1 ; TEXT ADDRESS:= TEXT ADDRESS + 1;

SN W0 10 ; IF CHAR = 10 THEN

JL. H4. ; GO TO END;

AL. W1 H1. ;

SE W0 63 ; IF CHAR = 63 THEN

JL. H3. ; CHAR:= 64;

AM 6 ;

J2: ;

H2: AL W0 58 ;

H H3: A50 , A31 ; WRITECHAR(CHAR); GO TO NEW;

J1. , A2+A23 ;

A50 , A31 ;

J1. , A2+A20 ;

W RL. W0 J1. ;

SH W0 64 ;

JL X1 ;

```
H4: JL.      JO,      ;
    AL W0     62      ; END:
    JL W1     H3.      ; WRITECHAR(BLACK RIBBON);
    RL W1     I1.      ;
    RL W2     I2.      ;
    A50      A39      ;
H      1<0     ; V(0)
W      0       ; END:
I0:     0       ;
I1:     0       ;
I2:     0       ;
J1:     0       ;
E
```

```

;      PROCEDURE CON CONVERTS A BINARY NUMBER
;      INTO AN ALPHANUMERIC STRING. IF THE NUMBER
;      OF THE CONVERTED CHARACTERS IS TOO BIG FOR THE
;      GIVEN PLACE, CON PUTS A - SIGN IN THE LAST
;      CHARACTER POSITION. THIS GIVEN PLACE OUGHT TO
;      BE OF EVEN CHARACTER LENGTH, OTHERWISE THE
;      LAST CHARACTER WILL ALWAYS BE A SPACE E.G.
;      X          NO OF CHARS    RESULT STRING
;      -1          3              , -1,
;      100         3              100,
;      1239        6              ,,1239
;      -123        2              2-

```

```

; WHERE , DENOTES A SPACE.

```

```

; PROCEDURE CON(NO OF CHARS, STRING ADD, X);
; REG          W0          X / DESTROYED
;              W1          SAVED
;              W2          SAVED
;              W3          RETURN / DESTROYED
; CALL          JL W3      (B24)
;              H <STRING ADD>, , <NO OF CHARS>

```

B H4, I7

```

H C2:      A50      ,      A38      ; BEGIN
W          1<2      ; P(2);

RS. W0      10.      ;
RS. W1      11.      ;
RS. W2      12.      ;
BL W1      X3        ;
WA W1      6          ; M:=STRING ADD;
AL W3      X3+2      ;
RS. W3      13.      ;
RS. W1      14.      ;
BZ W2      X3-1      ;
SH W0      -1        ;
AC W0      (0)       ; K:= ABS(X);
AL W1      8          ; L:= 8;
H1: AL W1      X1-1    ; REPEAT:
AL W3      0          ; L:= L-1;
WD. W0      J5.      ; DIGIT[L]:= IF K REM 10 = 0 THEN
SN W3      0          ; 16 ELSE K REM 10;
AL W3      16         ; K:= K : 10;
HS. W3      X1+16.    ;
SE W0      0          ; IF K = 0 THEN
JL.         H1.      ; GO TO REPEAT;
AL W3      32         ;
SH. W0      (10.)    ; IF X < 0 THEN
JL.         H2.      ; BEGIN
AL W1      X1-1      ; L:= L - 1;

```

```

      HS. W3      X1+16.      ;      DIGIT[L]:= < - >
H2:   AL  W0      32          ;      END;
      AC  W3      X2-8        ;      IF -, (L + 1) - NO OF CHARS + 8) THEN
      SL  W3      X1+1        ;
      HS. W0      17.         ;      DIGIT[7]:= < - >;
      HS. W1      15.         ;
      RL. W2      14.         ;      FOR K:= - NO OF CHARS + 8 STEP 1
H3:   AL  W0      0           ;      UNTIL 7 DO
H     SL  W3      , 15: 0      ;      BEGIN
W     BZ. W0      X3+16.      ;      CHAR:= IF K < L THEN 0 ELSE DIGIT[K];
      JL. W1      J6.         ;      OUTCH(CHAR);
      AL  W3      X3+1        ;
      SE  W3      8           ;
      JL.         H3.         ;      END;
      RL. W1      11.         ;
      RL. W2      12.         ;
H     A50      ,  A39         ;      V(2)
W     1<2        ;      END;

I3:   0           ;
I0:   0           ;
I1:   0           ;
I2:   0           ;
I4:   0           ;
J5:   10          ;
I6:   0           ;      INTEGER ARRAY DIGIT[0:7];
      0           ;
      0           ;
H     0           ;      , 17: 0
E

```

B H0
W J6:

SH W2 -1
JL. H0.
LS W0 6
HS W0 X2
AC W2 X2
JL X1
H0: AC W2 X2-1
BA W0 X2-1
HS W0 X2-1
JL X1

```
; PROCEDURE OUTCH(CHAR);
; IF M>0 THEN
; BEGIN
; STORE[M]:=CHAR SHIFT 6;
; M:=-M;
; END
; ELSE
; BEGIN
; M:=- (M-1);
; STORE[M-1]:=STORE[M-1]+CHAR
; END;
```

E

```

;   PROCEDURE INA INPUTS AN ALPHANUMERIC
;   TEXTSTRING OF GIVEN LENGTH. THE INPUT IS
;   ALWAYS FINISHED BY TYPING A TERMINATOR ; ,
;   / OR , . IF THE STRING (NOT INCLUDED THE
;   TERMINATOR) IS SHORTER THAN THE GIVEN LENGTH,
;   THE REMAINING POSITIONS WILL BE FILLED WITH
;   SPACES, OTHERWISE THE OVERFLOWING CHARACTERS
;   WILL BE SKIPPED.
;   CARRIAGE RETURNS, SPACES AND CASE SHIFTS
;   ARE SKIPPED DURING INPUT. HOWEVER, CASE
;   SHIFTS HAVE THE EFFECT ON SELECTING TERMINATOR

```

B 15

```

W   ; PROCEDURE INA(String ADDRESS, NO OF CHARS);
;   REG      W0      - / DESTROYED
;           W1      - / TERMINATOR x LAST CASE
;           W2      SAVED
;           W3      LINK
;   CALL     JL W3 (B25)
;           H <String ADDRESS>, , <NO OF CHARS>

```

B H2

```

H C3:   A50      , A38      ; BEGIN
W       1<1+1<0    ; P(1, 0);

RS. W2      15.      ;
BL W2      X3      ;
WA W2      6      ; M:=String ADDRESS;
AL W3      X3+2      ;
RS. W3      10.      ;
JL. W1      J2.      ; WRITECHAR(LC);
AL W0      58      ;
RS. W0      11.      ; CASE:= LC;
BZ W3      X3-1      ; FOR K:=M STEP 1
HS. W3      H1.      ; UNTIL M+NO OF CHARS:2 DO
AL W1      X2      ; STORE[K]:=0;
AL W0      0      ;
H0: HS W0      X1      ;
AL W1      X1+1      ;
AL W3      X3-2      ;
SL W3      1      ;
JL.      H0.      ;
H2: JL. W3      J3.      ; MORE: K:=INCH;
H AL W3      , H1: 0      ; NO OF CHARS:=NO OF CHARS-1;
W AL W3      X3-1      ;
HS. W3      H1.      ;
RL. W0      12.      ;
SL W3      0      ; IF NO OF CHARS>0 THEN
JL. W1      J6.      ; OUTCH(K);
JL.      H2.      ; GO TO MORE;

```

; TAPE 1 PART 2

12 05 1967

PAGE 11.

E

; END;


```

;      PROCEDURE IN ACCEPTS AN <INTEGER> NUMBER
; FROM THE TYPEWRITER. BLINDS AND TERMINATORS
; ARE THE SAME AS IN INA. THE NUMBER CAN
; CONSIST OF DIGITS AND + OR - SIGN. CHARACTERS
; NOT MENTIONED, ACT LIKE TERMINATOR, BUT
; IN THAT CASE THE TERMINATOR VALUE IN W1
; WILL BE COMPLEMENTED. (SEE REG) TYPING A
; TERMINATOR AS THE FIRST CHARACTER WILL
; RESULT THE NUMBER ZERO
;      <UNSIGNED INTEGER> ::= <DIGIT> <UNSIGNED INTEGER> <DIGIT>
;      <INTEGER> ::= <UNSIGNED INTEGER> + <UNSIGNED INTEGER>
;                      - <UNSIGNED INTEGER>

```

B H0

```

; INTEGER PROCEDURE IN;
; REG      W0      - / IN
;          W1      - / TERMINATOR * LAST CASE
;          W2      SAVED
;          W3      LINK
; CALL JL W3 (B27)

```

```

H C4:      A50      ,      A38      ; BEGIN
W          1<1+1<0    ; P(1, 0);
          RS. W2      I5.      ;
          RS. W3      I0.      ;
          JL. W1      J2.      ; WRITECHAR(LC);
          AL W0      58      ;
          RS. W0      I1.      ; CASE := LC;
          AL W2      0      ; NUMB := 0;
          HS. W2      I4.      ; MINUS := FALSE;
H0:        ; DIGIT:
          JL. W3      J3.      ; K := INCH;
          AL W3      -1      ;
          SN W1      1856      ; IF K = < - > THEN
          HS. W3      I4.      ; MINUS := TRUE;
          SN W1      928      ; IF K = < 0 > THEN
          RS. W0      I2.      ; CHAR := 0;
          SE W1      1856      ;
          SN W1      1920      ; IF K = < - > v K = < + > THEN
          JL.        H0.      ; GO TO DIGIT;
          RL. W0      I1.      ;
          SE W1      928      ;
          AC W1      X1      ; K := - K;
          SL W1      -522      ;
          SE W0      58      ; IF K < -9 x LC v CASE = LC THEN
          JL.        J4.      ; GO TO TERMINATED;
          WM. W2      J5.      ;
          WA. W2      I2.      ; NUMB := 10 x NUMB + CHAR;
          JL.        H0.      ; GOTO DIGIT;

```

E

; END;

```

; INTEGER PROCEDURE INCH;
; REG      W0      - / ZERO
;          W1      - / INCH
;          W2      SAVED
;          W3      LINK
; CALL     JL. W3 J3.
; INCH INPUTS ONE CHARACTER FROM THE
; TYPEWRITER, AND TREATS THE CASE, BLIND,
; AND TERMINATOR SYMBOLS.

```

```

B H3      ; BEGIN
W H0: RS. W1      I1. ; INCHAR:
H J3:      A50      , A31 ;
          I2.      , A2+A22 ;
          A50      , A31 ;
          I2.      , A2+A20 ;
W          RL. W1      I2. ; K:= CHAR FROM TYPEWRITER;
          SO. W1      (H1.) ; IF WATCH DOG ALARM THEN
          JL.          H2. ; BEGIN
H          A50      , A40 ;
W          1<0      ; V(0);
H          A50      , A37 ; PASS CONTROL;
          A50      , A38 ;
W          1<0      ; P(0);
          JL.          J3. ; GO TO INCHAR
          ; END;
H2: SH W1      64 ; IF ERROR THEN
          JL.      H3. ; BEGIN
          JL. W1      J0. ; SET ALARM LAMP;
          JL.      J3. ; GO TO INCHAR
          ; END;
H3: SE W1      58 ; IF K = 58 ~ K = 60 THEN
          SN W1      60 ; BEGIN CASE:= K;
          JL.      H0. ; GO TO INCHAR END;
          SE W1      0 ;
          SN W1      64 ; IF K = 0 ~ K = 64 THEN
          JL.      J3. ; GO TO INCHAR;
          WM. W1      I1. ; K:= K x CASE;
          SE W1      300 ; IF K = < ; > ~
          SN W1      1566 ; K = < , > ~
          JL.      J4. ; K = < / > THEN
          SE W1      180 ; GO TO TERMINATED
          JL          X3 ; END INCH;
J4:      ; TERMINATED:
          AL W0      X2 ; IN:= NUMB;
H          SO W0      , I4: 0 ; IF MINUS THEN
W          AC W0      X2 ; IN:= -NUMB;
H          RL. W2      I5. ;
          A50      , A39 ;

```

```
W          1<1+1<0 ; V(1, 0)
I0:         0 ; END;
I1:         0 ; INTEGER CASE,
I2:         0 ; CHAR;
I5:         0 ;
H1:        1<22
```

E
E
E

```

; PROCEDURE DEVICE ERROR MESSAGE;
;
; REG W0: TERMINAL NUMBER / DESTROYED
;      W1: DEVICE NUMBER < 6 / DESTROYED
;      W2: SAVED
;      W3: LINK
; CALL JL W3 (B21)
; COMMENT ERROR MESSAGE IS GIVEN ON THE CONTROL
;          TYPEWRITER:
;          DEVICE DD,DDDD ERROR, HHMM
;          WHERE THE FIRST NUMBER IS THE DEVICE NUMBER
;          AND THE SECOND ONE IS A TERMINAL ADDRESS, THE
;          REST INDICATES THE TIME WHEN THE ERROR OCCURED.
;          CONTROL ROOM LAMP IS LIGHTENED;

```

```

B      J3, I4      ; BEGIN
H C5:  A50      ,   A38 ; P(ERROR MESS)
W      1<4      ;
      RS. W2      J3. ;
      RS. W3      J1. ;
      JL W3      (B24) ; CON(TERMINAL NUMBER);
H      I3.      ,   4   ;
W      BZ W0      3   ;
      LS W0      -6   ;
      JL W3      (B24) ; CON(DEVICE NUMBER);
H      I2.      ,   2   ;
W      RL W1      B38 ;
      JL W3      (B19) ; CLOCK TO HOUR MIN(CLOCK);
      0
      RS. W1      I4. ;
      AL. W0      I1. ; OUTA(MESSAGE);
      JL W3      (B26) ;
      AL W0      A63 ; SET ALARM LAMP;
      JL. W1      C7. ;
      RL. W2      J3. ;
H      A50      ,   A39 ; V(ERROR MESS);
W      1<4      ;
      J1: 0      ;
      J3: 0      ;
      I1: <<63>DEVICE<0>> ; STRING MESSAGE,
      I2: <DD,<58>>      ; DEVICE NUMBER,
      I3: <DDDD>      ; TERMINAL NUMBER;
      <<0>ERROR,<0>>      ;
      I4: <HHMM<63><10>> ; END;

```

E

; PROCEDURE CLOCK TO HOUR MIN(W1);

;

; REG: W0: -

; W1: CLOCK / <HHMM>

; W2: -

; W3: LINK

; CALL: JL W3 (B19)

; <WORKING LOCATION>

; RETURN:

;

; COMMENT CLOCK AS BINARY SECONDS IS TRANSFOR-

; MED TO HOURS AND MINUTES AS CHARACTERS. 0=<16>;

```

B      H1, I2
W C6:  AL, W2      I1.  ;
H1:    AL W0      0    ;
      WD W1      X2    ;
      SN W0      0    ;
      AL W0      16    ;
      RX W1      X3    ;
      LD W1      -6    ;
      SN, W2     I2.   ;
      JL        X3+2   ;
      RX W1      X3    ;
      AL W2      X2-2   ;
      JL.        H1.   ;
      I2:      10      ;
           10          ;
           6           ;
           10          ;
      I1:      60      ;

```

E

```
; TAIL SEGMENT 1:
B E3
W E0: AL. W1      C8. ; UNPROTECT(
E1: PC          X1  ; B TABLE,
    AL W1      X1+2 ;
    SE. W1      C9. ;
    JL.         E1. ;
    AL. W1      A46. ;
E2: PC          X1  ; NAME TAB,
    AL W1      X1+2 ; ADDR TAB,
    SE. W1 A49,+2xA60 ; NEXT EXECUTION,
    JL.         E2. ; PERIOD);
    LD W2      48 ;
C [A54>22^1]-1
W JL W3      (B27) ;
E
    AL. W0      E0. ;
    JE W3      (B39) ; CALL LOADER;
    JL.         E0. ;

E L ; IDLIST S1
N T
E
L
; END S1
N
T
```


; THE OPERATORS CONTROL PROGRAM GIVES POSSIBILITY TO
; THE OPERATOR DOING SOME JOBS MANUALLY, AS FOLLOWS:

; 1. INPUT-OUTPUT OF THE INTERNAL TIMER
; 2. INPUT OF THE DATE
; 3. I-O HIGH LIMITS IN THE SCAN TABLE
; 4. I-O LOW - - - -
; 5. I-O OTHER PARAMETERS - -
; 6. INSERTING RANGE
; 7. REMOVING -
; 8. I-O OF THE PLANT DATA
; 9. CONTROLLING TREND-LOG
; 10. STARTING TASK PROGRAM IN SINGLE MODE
; 11. I-O ACTIVE AND PASSIVE GROUPS
; 12. I-O NEXT EXECUTION TIME OF TASK PROGRAMS
; 13. I-O PERIOD - - - -
; 14. CLOSING TASK PROGRAMS
; 15. LOADING BINARY TAPE
; 16. RESET
; 17. I-O SPECIAL RUN MODES
; 18. LISTING PAPER TAPE ON THE CONTROL TYPEWRITER
; 19. STARTING MULTISYSTEM

; TEST OUTPUT: NONE.

; SETTING BIT IN A56 CAUSES OVERRIDING ERRORACTION IN JOB 5.

; SEGMENT NUMBER : 3

```

C      [A55>20^1] - 1
S      F26,G10,H80,I5,J5      ; SEGMENT 3:
K=0
W
H1:   AL. W0      F2,      ; ERROR:
      JL. W3      (B26)    ;   WRITETEXT(< />);
H2:   AL. W0      F3,      ; CR:
      JL. W3      (B26)    ;   WRITECR;
H30:
H3:   JL. W3      (B25)    ;   READY:
      JL. W3      (B25)    ;   INTEXT(I) NO OF CHARS: (4);
H      I0,      4      ;
W      JL. W3      H46,    ;   TEST TERM;
      RL. W0      I0,      ;   FOR K:= 0 STEP 1 UNTIL NO OF JOBS DO
      AL. W1      -2      ;   IF JOBS[K] = 1 THEN
H5:   AL. W1      X1+2    ;   GO TO FOUND;
      SN. W1      J2      ;
      JL.      H1,      ;   GO TO ERROR
      SE. W0      (X1+J1.) ;
      JL.      H5,      ;   FOUND:
      LS. W1      -1      ;
      BL. W1      X1+J3.   ;
H6:   JL.      X1+H6.    ;   GO TO JOB[K+1];

H10:  AL. W3      B38     ; CLOCK:
H52:  RS. W3      I0,     ;   I:= ADDRESS OF CLOCK;
      AL. W0      F12,    ; TIME:
      JL. W3      (B26)   ;   WRITETEXT(<TIME>);
      AL. W2      0       ;   SEC:= 0;
H8:   JL. W3      (B27)   ; NEXT:
      AL. W3      -1      ;   K:= INONE;
      SN. W1      (G5,)   ;   IF TERM = <H> THEN
      WM. W0      G0,     ;   K:= K x 3600;
      SN. W1      (G6,)   ;   IF TERM = <M> THEN
      WM. W0      G1,     ;   K:= K x 60;
      SN. W1      -1044   ;   IF TERM = <S> ^
      AL. W3      0       ;   TERM = <M> ^
      SE. W3      0       ;   TERM = <H> THEN
      JL.      H7,      ;   GO TO TERMINATED;
      WA. W2      0       ;   SEC:= SEC + K;
      JL.      H8,      ;   GO TO NEXT;
H7:   ; TERMINATED:
      JL. W3      H46,    ;   TEST TERM;
      SE. W1      -240    ;   IF TERM = < = > THEN
      JL.      H9,      ;   BEGIN
      RL. W1      (I0,)   ;
      SE. W1      (G3,)   ;
      JL.      6         ;
      AL. W0      F26,    ;
      JL.      H3, -2    ;

```

```

      AL W2      F21 ;
H75: AL W0      0 ; WRITETEXT(
      WD. W1     G1. ; CONVERT CLOCK(STORE[I]);
      HS. W2     H76. ;
      JL W3      (B24) ;
H H76: 0 , 2 ;
W      AL W2     X2-2 ;
      SE W2     F22 ;
      JL.       H75. ;
      AL. W0     F5. ;
      JL W3      (B26) ;
      JL.       H3. ; GO TO READY
                        ; END;
H9: SE W0      0 ; IF K = 0 THEN
      JL.       H1. ; GO TO ERROR;
      RS. W2     (I0.) ; STORE[I]:= SEC;
      JL.       H30. ; GO TO READY;

                        ;
G0:      3600 ; CONSTANTS
G1:      60 ; SECS IN HOUR
G5:      -3248 ; SECS IN MINUTE
G6:      -2088 ; -( <H> x LC)
G7:      1<20 ; -( <M> x LC)
G2:      -1-32 ; BIT NO PAPER
G3:      1<22 ; -, BIT R
H11: AL. W0     F17. ; INPUT DATE;
      JL W3      (B26) ; WRITETEXT(<DAY>);
      JL W3      (B27) ; K:= INONE;
      JL. W3     H46. ; TEST TERM;
      JL W3      (B24) ; DAY:= CONVERT(K);
H      F23. , 2 ;
W      AL. W0     F18. ; WRITETEXT(<MONTH, YEAR>);
      JL W3      (B26) ;
      JL W3      (B25) ; INTEXT(REST) NO OF CHARS: (6);
H      F24. , 6 ;
W      JL. W3     H46. ; TEST TERM;
      RL. W0     F23. ;
      RS W0      B41 ;
      RL. W0     F25. ;
      RS W0      B41+2 ;
      JL.       H3. ; GO TO READY;
W
H F23: 0 , F24: 0 ; <DDMM
W F25:
I0:      0 ; YYYY>
H18: ; TREND:
      AM      64 ; TREND:= TRUE; GO TO SCAN;

```

```

H15:                                ; INSERT:
      AM                            -32 ;   INSERT:= TRUE; GO TO SCAN;
H16:                                ; REMOVE:
      AM                            28  ;   REMOVE:= TRUE; GO TO SCAN;
H12:                                ; HIGH:
      AM                            2   ;   HIGH:= TRUE; GO TO SCAN;
H13:                                ; LOW:
      AM                            1   ;   LOW:= TRUE; GO TO SCAN;
H14:                                ; PARAM:
      AL W0                          1   ;   PARAM:= TRUE;
      RS W0                          10. ;   SCAN:
      AL W0                          F6. ;
      JL W3                          (B26) ;   WRITETEXT(<SCAN NO>);
      JL W3                          (B27) ;   SCAN NO:= INONE;
      JL W3                          H46. ;   TEST TERM;
      AL W3                          0   ;   COUNT:= LAST ENTRY:= 0;
      HS W3                          H33. ;
H32: JL W2                          (B23) ; NEXT ENTRY:
      SH W0 1001111111111111 ;   LAST ENTRY:= NEXT ENTRY (LAST ENTRY);
      JL W0                          H34. ;
      SH W0                          -1   ;   IF END THEN GO TO END;
      JL W0                          H32. ;   IF GROUP OR CR THEN GO TO NEXT ENTRY;
H    AL W2                          , H33: 0 ;
W    AL W2                          X2+1 ;   COUNT:= COUNT + 1;
      HS W2                          H33. ;
      SE W2                          (I3.) ;   IF COUNT = SCAN NO THEN
      JL W2                          H32. ;   GO TO NEXT ENTRY;
      RX W1                          10. ;   STORE NUMBER OF CONSTANTS;
      SO W1                          64   ;   IF TREND THEN
      JL W2                          H35. ;   BEGIN
      RS W2                          B42   ;   TREND NO:= SCAN NO;
      JL W2                          H3.   ;   GO TO READY
H35: AL W2                          X3+1 ;   END;
      SO W1                          2   ;   IF LOW THEN
      JL W2                          H41. ;   BEGIN
      SO W0                          256  ;   IF -, L THEN
      JL W2                          H1.   ;   GO TO ERROR;
      SO W0                          512  ;   IF -, H THEN
      JL W2                          H38. ;   GO TO PLUS ONE;
      AL W2                          X2+1 ;   TABLE ADD:= LAST ENTRY + 2;
      JL W2                          H38. ;   GO TO VALUE;
      ;                               ;   END;
H41: SO W1                          4   ;   IF HIGH THEN
      JL W2                          H36. ;   BEGIN
      SO W0                          512  ;   IF -, H THEN
      JL W2                          H1.   ;   GO TO ERROR;
      ;                               ; PLUS ONE:
      ;                               ;   TABLE ADD:= LAST ENTRY + 1;
      JL W2                          H38. ;   GO TO VALUE

```

```

; END;
H36: SO W1 1 ; IF -, PARAM THEN
      JL. H39. ; GO TO RANGE
      ; TABLE ADD:= LAST ENTRY + 1 +
      SZ W0 256 ; (IF L THEN
      AL W2 X2+1 ; 1 ELSE 0) +
      SZ W0 512 ; (IF H THEN
      AL W2 X2+1 ; 1 ELSE 0);
      ; A56<3 ;
C W LA. W0 H72. ;
E HS. W0 H73. ;
      ;
      AL. W0 F7. ;
      JL W3 (B26) ; WRITETEXT(<ITEM NO>);
      JL W3 (B27) ; K:= INONE;
      JL. W3 H46. ; TEST TERM;
      ; A56<3 ; IF -, SPECIAL REQUEST THEN
      ; H73: 0 ; BEGIN
      ; SL W0 1 ; IF K<1 ~
      ; SL W0 3 ; -, K<3 ~
      ; JL. H1. ;
      ; SN W0 2 ; ( IF K=2 THEN
      ; SH W1 3 ; EQ>3 ELSE
      ; SL W1 11 ; -, EQ<11 ) THEN
      ; JL. H1. ; GO TO ERROR
      ; END;
      ; WA W2 0 ; TABLE ADD:= TABLE ADD + K;
H38: ; VALUE:
      ; WA W2 B30 ; ABS ADD:= TABLE ADD + TABLE BASE;
      ; AL. W0 F4. ;
      ; JL W3 (B26) ; WRITETEXT(<VALUE>);
      ; JL W3 (B27) ; K:= INONE;
      ; JL. W3 H46. ; TEST TERM;
      ; SE W1 -240 ; IF TERM = < = > THEN
      ; JL. H40. ; BEGIN
      ; BZ W0 X2 ;
      ; JL W3 (B24) ;
      ; H W F8. , 4 ; OUTPUT(<NDDD;>, STORE[ABS ADD]);
      ; AL. W0 F8. ;
      ; JL W3 (B26) ;
      ; JL. H3. ; GO TO READY
      ; END;
      ; H40: HS W0 X2 ; STORE[ABS ADD]:= K;
      ; JL. H3. ; GO TO READY;

      ; H39: ; RANGE:
      ; WA W2 B30 ; ABS ADD:= LAST ENTRY + TABLE BASE;
      ; BZ W0 X2 -1 ; SET R BIT OF (STORE[ABS ADD])
      ; LA. W0 G2. ; TO: (INSERT ^ -, REMOVE);

```

```

      LO W0      2      ;
      HS W0      X2 -1   ; GO TO READY;
      JL.        H3.    ;
H34:  SO W0      2      ; END:
      JL.        H32.   ; IF TYPEPART = 2 THEN GO TO NEXT ENTRY;
      JL.        H1.    ; GOTO ERROR;

H17:                                ; PLANT DATA:
      AL W2      0      ; FOR K:= 0 STEP 2 UNTIL NO OF DATA - 2 DO
H45:  LS W2      -1     ; BEGIN
      BL W3      X2+J4.  ;
      LS W2      1      ;
H42:  AL W0      X3+H42. ;
      JL W3      (B26)   ; WRITETEXT(STORE[PLANT DATA[K]]);
      JL W3      (B27)   ; L:= INONE;
      JL W3      H46.    ; TEST TERM;
      AL W3      X2+E2   ; ABS ADD:= K + NO OF RESULTS +
      WA W3      B29     ; RESULT TABLE BASE;
      SE W1      -240    ; IF TERM = < = > THEN
      JL.        H43.    ; BEGIN
      RL W0      X3      ;
      JL W3      (B24)   ;
      F9.        8      ; OUTPUT(<-DDDDDDD;> STORE[ABSADD]);
      AL W0      F9.     ;
      JL W3      (B26)   ;
      JL.        H44.    ; END
H43:  RS W0      X3      ; ELSE STORE [ABS ADD]:= L
H44:  AL W2      X2+2    ;
      SE W2      E3      ;
      JL.        H45.    ; END;
      JL.        H3.     ; GO TO READY;

H46:  RS W0      I3.     ;
      SN W1      -240    ; PROCEDURE TEST TERM;
      JL.        X3      ; BEGIN
      SN W1      180     ; IF TERM = < / > THEN
      JL.        H2.     ; GO TO CR;
      SH W1      -1      ; IF TERM < 0 ^ TERM = < = > THEN
      JL.        H1.     ; GO TO ERROR;
      SE W1      300     ; IF TERM = < ; > THEN WRITECR
      JL.        X3      ; END;
      RS W3      I2.     ;
      AL W0      F3.     ;
      JL W3      (B26)   ;
      RL W0      I3.     ;
      JL.        (I2.)   ;
I2:   0          0      ;
I3:   0          0      ;

```

```

H26:                                     ; MULTI START:
      AM                               -1   ; K:= -2; GO TO TASK;
H19:                                     ; SINGLE START:
      AM                               -1   ; K:= -1; GO TO TASK;
H23:                                     ; CLOSE:
      AM                               -A49 ; K:= 0; GO TO TASK;
                                     ; PERIOD:
H22:
      AM                               A49-A48 ; K:= PERIOD TABLE BASE; GO TO TASK;
H21:                                     ; OPEN:
      AL W2                           A48   ; K:= NEXT EXECUTION TABLE BASE;
      AL W0                           F11.  ; TASK:
      JL W3                           (B26) ; WRITETEXT(<TASK>);
      JL W3                           (B25) ; INTEXT(I) NO OF CHARS: (4);
H      IO.                            4     ;
W      JL W3                           H46.  ; TEST TERM;
      RL W0                           IO.   ;
      RS W0                           A45   ; NAME[LAST]:= I;
      AL W3                           -2    ; FOR I:= 0, J+2 WHILE TRUE DO
H50: AL W3                           X3+2   ;
      SE W0                           (X3+A46) ; IF NAME[J] = I THEN GO TO OK;
      JL.                             H50.   ;
                                     ; OK:
      SN W3                           A45-A46 ; IF J = LAST THEN
      JL.                             H1.   ; GO TO ERROR;
      AL W0                           X3-1   ; IF K = -2 THEN
      WS W0                           4     ; MULTI START(J);
      SH W2                           -1    ; IF K = -1 THEN
H      A50 , A34                       ; SINGLE START(J);
W      SH W2                           -1    ; IF K<0 THEN
      JL.                             H3.   ; GO TO READY;
      SE W2                           0     ; IF K = 0 THEN
      JL.                             H51.  ; BEGIN
      RL W0                           G3.   ;
      RS W0                           X3+A48 ; NEXTEXECUTION[J]:= INFINITE;
      JL.                             H3.   ; GO TO READY
                                     ; END;
H51: WA W3                           4     ; I:= K + J;
      JL.                             H52.  ; GO TO TIME;

H24: AL W0                           F13.  ; RUN MODE:
      JL W3                           (B26) ; WRITETEXT(<TYPE>);
      AL W3                           B37   ; I:= RUN MODE ADDRESS;
      AL W1                           15    ; SHIFT:= 15;
      JL.                             H56.  ; GO TO TAKE NUMBER;

H20: AL W0                           F10.  ; GROUP:
      JL W3                           (B26) ; WRITETEXT(<GROUP NO>);
      AL W1                           23    ; SHIFT:= 23;

```

```

      AL W3      B36 ; I:= GROUP DESCRIPTION ADDRESS;
H56: HS. W1      H64. ;
      RS. W3      IO. ;
                                ; TAKE NUMBER:
H68: JL W3      (B27) ; K:= INONE;
      JL. W3      H46. ; TEXT TERM;
      SN W1      -240 ; IF TERM = < = > THEN
      JL.         H48. ; GO TO OUT;
      AC W3      (0) ;
      SH W0      -1 ;
      AC W3      X3 ;
      AL W2      1 ;
H    LS W2      X3 ; L:= 1 SHIFT (SHIFT - ABS(K));
      H64: 0 ;
W    RL. W3      (IO.) ; STORE[I]:=
      LO W3      4 ; IF K < 0 THEN
      AC W2      X2+1 ; STORE[I] ^ -, L ELSE
      SH W0      -1 ; STORE[I] v L;
      LA W3      4 ;
      RS. W3      (IO.) ;
      SN W1      1566 ; IF TERM = < , > THEN
      JL.         H68. ; GO TO TAKE NUMBER;
      JL.         H3. ; GO TO READY;
H48: ; OUT:
      RL. W1      (IO.) ; PATTERN:=
      AL W3      24 ; STORE[I]
      BS. W3      H64. ; SHIFT (24 - SHIFT)
      LS W1      X3 ;
H72: AL W2      31 ; COMMENT 31 IS USED AT H36;
H67: LD W1      1 ; FOR K := 1 STEP 1 UNTIL 15 DO
      SO W0      1 ; IF BIT(PATTERN, K) THEN
      JL.         H60. ;
H    AL W3      , <,<10>> ; OUTPUT(
W    SN W1      0 ; IF BITS(PATTERN, K+1,23) = 0 THEN
H    AL W3      , <<60><69>>; <ND;> ELSE <ND,>, K);
W    HS. W3      F19. ;
      AL W0      X2-30 ;
      JL W3      (B24) ;
H    F15. , 2 ;
W    AL. W0      F15. ;
      JL W3      (B26) ;
H60: AL W2      X2+1 ;
      SE W2      46 ;
      JL.         H67. ;
      JL.         H2. ; GO TO CR;

H29: ; RESET:
H    A50 , A33 ; RESET;

```



```

W H28: RL W0 B17 ; LOAD:
      JL W3 (B24) ; OUTPUT(<DDDD>, LAST);
H      F8. , 4 ;
W      AL W0 F8. ;
      JL W3 (B26) ;
      JL W3 (B27) ; K:= INONE;
      SE W1 (G8.) ; IF TERM = <P> THEN
      JL W3 H46. ; TEST TERM;
      AL W2 0 ; PROTECTION:= TERM = P;
      SN W0 0 ; IF K = 0 THEN
      RL W0 B17 ; K:= LAST;
      SN W1 -240 ; IF TERM = < = > THEN
      JL H1. ; GO TO ERROR;
      JL W3 (B39) ; CALL LOADER;
G8: -2262 ;

H27: ; LIST:
H      A50 , A31 ;
      IO. , A22+A0 ;
      A50 , A31 ;
      IO. , A20+A0 ; I:= INCHAR
W      RL W0 IO. ;
      SO W0 (G7.) ; IF NO PAPER THEN
      JL H69. ; BEGIN
      AL W0 F16. ;
      JL W3 (B26) ; WRITETEXT(<NO PAPER>);
      JL H3. ; GO TO READY
      ; END;
H69: SL W0 65 ; IF ERROR THEN
      JL H70. ; GO TO READER;
H      A50 , A38 ;
      IO. , 1<0 ; P(TW);
H      A50 , A31 ;
      IO. , A23+A2 ;
      A50 , A31 ;
      IO. , A20+A2 ; WRITECHAR(I)
      A50 , A40 ; V(TW);
      IO. , 1<0 ;
      A50 , A37 ; PASS CONTROL;
W      RL W0 IO. ;
      SH W0 64 ; IF -, ERROR THEN
      JL H27. ; GO TO LIST;
      AM A2-A0 ; DEVICE:=TW; GO TO ERROR;
H70: ; READER:
      AL W1 A0 ; DEVICE:= READER;
      AL W0 0 ; ERROR:
      JL W3 (B21) ; ERRORMESSAGE(0, DEVICE);
      JL H2. ; GO TO CR;

```

B E6

; ACTIONNAMES:

W J1: <CLOC> , <DATE>
 <HIGH> , <LOWL>
 <PARA> , <INSE>
 <REMO> , <PLAN>
 <TREN> , <SING>
 <GROU> , <OPEN>
 <PERI> , <CLOS>
 <LOAD> , <RESE>
 <RUNM> , <LIST>
 <MULT>

H J3: J2 = K - J1

H10-H6 , H11-H6
 H12-H6 , H13-H6
 H14-H6 , H15-H6
 H16-H6 , H17-H6
 H18-H6 , H19-H6
 H20-H6 , H21-H6
 H22-H6 , H23-H6
 H28-H6 , H29-H6
 H24-H6 , H27-H6
 H26-H6

; DEFINE NO OF JOBS
 ; SWITCH JOB:=
 ; CLOCK, INPUT DATE
 ; HIGH, LOW
 ; PARAM, INSERT
 ; REMOVE, PLANT DATA
 ; TREND, SINGE START
 ; PASSIVE, OPEN
 ; PERIOD, CLOSE
 ; LOAD, RESET
 ; RUNMODE, LIST
 ;MULTISTART;

; PLANT DATA NAMES

E1: <HEAT <10>>
 E2: <PURGE <10>>
 E3: <LETDOWN<10>>
 E4: <WTPRC <10>>
 E5: <RHO <10>>
 E6: <INLET <10>>

J4: E1-H42 , E2-H42
 E3-H42 , E4-H42
 E5-H42 , E6-H42

; USED TEXTS:

H F4: <VALUE<10>>
 F6: <SCAN NO<10>>
 F7: <ITEM NO<10>>
 F11: <TASK <10>>
 F12: <TIME <10>>
 F10: <NO <10>>
 F13: <TYPE <10>>
 F16: <NO PAPER<63><10>>
 F17: <DAY<10>>
 F18: <MONTH, YEAR<10>>

F2: </<10>>

F3: <<63><10>>

; CLOCK LAYOUT:

F5: <DDH<0>DDM<0>DDS;<63><10>>

F21= F5+4-H76

F22= F5-2-H76

; LAYOUTS:

F9: <-DDD>

F8: <DDDD;<63><10>>

F15: <DD>

F19: <,<10><58><10>>

F26: <CLOSED<10>>

E
W

```
B          E1          ; TAIL SEGMENT 3:
W E0:  RL. W0          E1. ;
      RS. W0          A46+0 ;
      AL. W0          H3.  ; SET USER(0, READY,<OPER>);
      RS. W0          A47+0 ;
      AL. W0          E0.  ;
      AL. W1          0    ;
      AL. W2          A54>20^1 ;
      JL. W3          (B39) ; CALL LOADER;
E1:      <OPER>
      JL.          E0.  ;

E
L
; IDLIST S3
N
T
E
E
L
; END S3
N
T
```


C [A55>19^1] - 1
S C20, D10, F13
W K=0

; SEGMENT 4:

;

; THE SEGMENT CONTAINS THE FOLLOWING TASK

; PROGRAMS:

; PULSE INTEGRATION

; FLOW INTEGRATION

; BALANCE CALCULATION RESET AND PRINTING PHASE

; LOG 1

; LOG 2

; ALARM SCAN

; TREND LOG

```
; PULSE SCAN:
;
; COMMENT:
; THE DIGITAL PULSE REGISTERS ARE INPUT AND STORED
; LOCALLY IN THE PULSE WORD TABLE.
;
; GUIDED BY THE PULSE SCAN TABLE VALUES IN THE PULSE
; INTEGRATION TABLE ARE COUNTED UP BY ONE DEPENDING
; ON CORRESPONDING BITS IN THE PULSE WORD TABLE.
;
; THE ORDER OF ENTRIES IN THE PULSE SCAN TABLE, AND
; THE PULSE INTEGRATION TABLE IS THE SAME.
;
; FOLLOWING ACTIONS IS TAKEN ON ENTRIES IN THE PULSE
; SCAN TABLE:
; GROUP ENTRIES: A BOOLEAN IS SET TELLING WHETHER
;   CURRENT GROUP IS ACTIVE OR PASSIVE.
; CR ENTRIES: THE ENTRIES ARE NEGLECTED.
; PULSE ENTRIES: A POINTER, POINTING AT AN ENTRY
;   IN THE PULSE INTEGRATION TABLE IS COUNTED UP
;   BY ONE.
;   THE PULSE ENTRY CONTAINS A WORD NUMBER AND
;   A BIT NUMBER REFERRING TO A BIT IN THE PULSE
;   WORD TABLE.
;   IF CURRENT GROUP IS ACTIVE AND THE CORRESPON-
;   DING BIT IN THE PULSE WORD TABLE IS ONE, TWO
;   IS ADDED TO THE ENTRY POINTED AT IN THE
;   PULSE INTEGRATION TABLE.
; END ENTRY: THE SCANNING IS TERMINATED.
;
; NORMAL PERIOD: 1 SEC.
;
; EXECUTION TIME: PER PULSE ENTRY: 0.07 MS
;                  FOR CA 70 ENTRIES : 5 MS
;
```

```

B      H3, J1, I3      ; BEGIN
H C1:      A50      ,      A38      ; P(PULSE BAL);
W          1<12      ;
          AL W1      A7      ; FOR I:= 0 STEP 1 UNTIL 2 DO
          AL W3      I1,      ; PULSE WORD[I]:=
          JL W2      C13,      ; DIG VALUE(PULSE DEVICE[I]);
          RS W0      X3      ;
          AL W1      X1+1<6-A24;
          AL W3      X3+2      ;
          SH W3      I3,      ;
          JL          C13.      ;

          RL W3      B33      ; ENTRY 1:= PULSE SCAN TAB FIRST ENTRY-1;
          RL W2      B35      ; ENTRY 2:= PULSE INTEGER TAB FIRST ENTRY;
H1:      BZ W0      X3      ; STEP:
          AL W3      X3+1      ; ENTRY 1:= ENTRY 1 + 1;
          LD W1      -4      ; W0:= PULSE SCAN TAB[ENTRY, TYPE AND BIT NO.
          LS W1      -20     ; W1:= PULSE SCAN TAB [ENTRY, WORD NO];
          SZ W0      10 000000 ; IF TYPE = PULSE ENTRY
          JL          H2,      ; THEN GO TO TEST GROUP;
          AL W2      X2+2      ; ENTRY 2:= ENTRY 2 + 2;
          RL W1      X1+I1.      ;
          LS W1      (0)      ;
H      AL W0      , J1: 0      ; BOOLEAN GROUP ACTIVE:
W      SH W0      -1      ; IF PULSE WORD[WORD NO, BIT NO]
          SL W1      0      ; ^ GROUP ACTIVE
          JL          H1,      ; THEN
          RL W1      X2-2      ; PULSE INTGR TAB[ENTRY 2 - 2];=
          AL W1      X1+1      ; PULSE INTGR TAB[ENTRY 2 - 2] + 1;
          RS W1      X2-2      ;
          JL          H1,      ; GO TO STEP;

H2:      SO W0      00 100000 ; TEST GROUP:
          JL          H3,      ; IF TYPE = GROUP THEN GO TO TEST END 5;
          RL W0      B36      ;
          LS W0      X1-12     ; GROUP ACTIVE:= GROUP [W1];
          HS W0      J1,      ;
          JL          H1,      ; GO TO STEP;
H3:      SO W0      00 000001 ; TEST END 5:
          JL          H1,      ; IF TYPE = END 5 THEN GO TO STEP;
H      A50      ,      A40      ; V(PULSE BAL);
W          1<12      ;
H      A50      ,      A32      ; EXIT PROGRAM;
W I1: 0, I2: 0, I3: 0      ; BOOLEAN ARRAY PULSE WORD[0:2, 1:23];
E          ; END PULSE SCAN;

```


; FLOW INTEGRATION:

; THE PROGRAM TAKES THE ITEMS IN THE SCAN
; TABLE, AND EXECUTES AN ACTION DEPENDING
; ON THEIR TYPE:

; 1. A/D ENTRY WHICH HAS AN I (INTEGRATION) BIT:

; THE INTEGRATION TABLE INDEX IS INCREASED BY
; ONE, IF THE ENTRY IS NOT REMOVED (NO R BIT)
; AND THE GROUP IS ACTIVE THE VALUE CORRESPONDING
; TO THE ENTRY WILL BE CALCULATED, AND ADDED TO
; THE INTEGRATION TABLE ITEM, POINTED OUT BY THE
; INDEX.

; 2. GROUP:

; THE PROGRAM SETS THE BOOLEAN, WHETHER THE
; GROUP IS ACTIVE OR NOT

; 3. END 2:

; EXIT FROM THE PROGRAM IS MADE

; 4. ALL OTHER ITEMS ARE TREATED AS DUMMIES

; THE PROGRAM HAS A SEMAPHOR TO PREVENT A
; BALANCE CALCULATION WHILE RUNNING, AND ALSO COUNTS,
; THE NUMBER OF RUNS.

; NORMAL PERIOD: 5 MIN

;

```

B      H4, I2      ; BEGIN
C2:    ; COMMENT FLOW SCAN;
H      A50      ,   A38      ;
W      1<11      ; P(FLOW BAL);
      RL W3      B40      ;
      AL W3      X3+1      ;
      RS W3      B40      ; NO OF STEPS:= NO OF STEPS + 1;
      RL W3      B34      ;
      RS W3      I0.      ; I:= FLOW TABLE BASE; LAST:= 0;
      AL W3      0      ; NEXT:
H1:    JL W2      (B23)      ; LAST:= NEXT ENTRY(LAST);
H      AL W2      , I1: 0      ; BOOLEAN ACTIVE;
W      SH W0      -1      ; IF AD ENTRY THEN
      JL.      H2.      ; BEGIN
      SO W0      10 00000      ; IF -, I BIT THEN
      JL.      H1.      ; GO TO NEXT;
      RS W3      I2.      ;
      RL W1      I0.      ;
      SH W2      -1      ; IF ACTIVE ^ -, R BIT THEN
      SZ W0      01 00000      ;
      JL.      H3.      ; FLOWTABLE[I]:= FLOWTABLE[I] +
      JL W2      (B20)      ; VALUE(LAST);
      RL W1      I0.      ;
      WA W0      X1      ;
      RS W0      X1      ;
H3:    AL W1      X1+2      ; I:= I + 2;
      RS W1      I0.      ;
      RL W3      I2.      ;
      JL.      H1.      ; GO TO NEXT
      ; END;
H2:    SO W0      512      ; IF -, GROUP THEN
      JL.      H4.      ; GO TO PERHAPS END;
      LD W1      -4      ;
      LS W1      -20      ;
      RL W0      B36      ; ACTIVE:= GROUP DESC[GROUP NO];
      LS W0      X1-12      ;
      HS W0      I1.      ;
      JL.      H1.      ; GO TO NEXT;
H4:    ; PERHAPS END:
      SO W0      2      ; IF TYPE = 2 THEN
      JL.      H1.      ; GO TO NEXT;
      ;
H      A50      ,   A40      ;
W      1<11      ; V(FLOW BAL);
H      A50      ,   A32      ; EXIT PROGRAM;
W
I0:    0      ; INTEGER I;
I2:    0      ; SAVE LAST;
E      ; END;

```


; BALANCE PROGRAM

; THE PROGRAM EXECUTES THREE PHASES:

; 1. CALCULATION PHASE:

; A PROGRAM FOR PERFORMING THE BALANCE CALCULATION
; IS CALLED. THE RESULTS OF THE CALCULATION ARE STORED
; IN A TABLE, THE RESULT TABLE, IN AN ORDER
; IN WHICH THE RESULTS SHALL BE PRINTED.

; 2. RESET PHASE:

; THE ENTRIES IN THE PULSE INTEGRATION TABLE,
; AND THE FLOW INTEGRATION TABLE ARE RESET TO ZERO.

; DURING PHASE 1 AND 2 SEMAPHORES ARE SET SO THAT
; NO PULSE INTEGRATION AND NO FLOW INTEGRATION WILL
; TAKE PLACE DURING CALCULATION AND RESETTING.

; 3. PRINT PHASE:

; THE RESULTS COLLECTED IN THE RESULT TABLE ARE
; PRINTED. THE FORMAT FOR PRINTING IS GIVEN
; BY INFORMATION STORED IN A TABLE, THE LINE
; TABLE, WHICH IS LOCAL TO THIS PROGRAM.

; BY RUNMODE CAN BE SPECIFIED:

; A: EXECUTION OF CALCULATION PHASE
; B: EXECUTION OF RESET PHASE
; C: PRINTING ON LOG TWR 1
; D: PRINTING ON LOG TWR 2
; E: PRINTING ON PUNCH.

; NORMAL RUNMODE WILL BE: A, B, C, AND D.

; NORMAL PERIOD: 8 HOURS

```

B      H14, I16, J10      ; BEGIN
;
H C3:      A50      ,      A38      ; P(FLOW BAL, PULSE BAL);
W      1<12+1<11      ;

      RL W1      B37      ; CALCULATION PHASE:
      SO W1      1<10      ; IF RUNMODE=BAL CALCULATION
      JL.      C17.      ; THEN GO TO RESET PHASE;
      RL W2      B38      ;
      RS. W2      I16.      ; LAST CALC:=CLOCK;
      WS W2      B45      ; PERIODE:=CLOCK-LAST RESET;
      SH W2      0      ; IF PERIODE<0 THEN
      WA. W2      D2.      ; PERIODE:=PERIODE+DAY IN SECONDS;
      RS W2      B46      ;
      JL      (B44)      ; BALANCE CALCULATION;
C17: RL W1      B37      ; RESET PHASE:
      AL W0      0      ;
      SO W1      1<9      ; IF RUNMODE = BAL RESET
      JL.      H2.      ; THEN
; BEGIN
      RS W0      B40      ; NO OF STEPS:=0;
      RL W2      B34      ; FOR I:= INTEGR TAB LAST WORD,
      AL W2      X2+E0+E1      ; I-2
H1: AL W2      X2-2      ; WHILE I > INTEGR TAB FIRST WORD
      RS W0      X2      ; DO
      SE W2      (B34)      ; INTEGR TAB[I]:= 0;
      JL.      H1.      ;
      RL W2      B38      ;
      RS W2      B45      ; LAST RESET:=CLOCK;
H H2: A50      ,      A40      ; V(FLOW BAL, PULSE BAL);
W      1<12+1<11      ; END;
; PRINT PHASE:
      LS W1      6      ; LOG SEMAPHORES:=
      LA. W1      D5.      ; RUNMODE SHIFT 3 ^ 111 < 6;
      RS. W1      J1.      ;
      RS. W1      J5.      ;
H      A50      ,      A38      ; P(LOG SEMAPHORES);
W J1:      0      ;
      HS. W0      F1.      ; LOG 1 CR:= 0;
      HS. W0      F2.      ; LOG 2 CR:= 0;

; PRINT HEADING:
      RL. W1      I16.      ;
      JL W3      (B19)      ; CLOCK STRING:=
      0      ; CLOCK TO HOUR MIN(LAST CALC);
      RS. W1      I1.      ;
      AL. W3      I2.      ;
      JL. W2      C12.      ; TRANSFER DATE (DATE STRING);
      RL W1      B46      ;
      JL W3      (B19)      ; PERIODE STRING:=
      0      ; CLOCK TO HOUR MIN (PERIODE);

```

```

RS. W1          I3. ;
AL. W3          I4. ;
JL. W2          H8. ;   OUT(HEADING);
                        ; PRINT RESULTS:
RL W3          B29 ;   POINTER[1]:=RESULT TABLE BASE;
RS. W3          I10. ;
AL W3          X3+E9 ;   POINTER[2]:=POINTER[1]+
RS. W3          I11. ;   FIRST RESULT ON THE RIGHT SIDE;
AL W3          -1 ;   INDEX:=-1;
                        ; NEXT LINE:
H3: AL W3      X3+1 ;   INDEX:=INDEX+1;
    HS. W3      H4. ;
    BZ. W2      X3+D1. ;
    SN W3      H14 ;   IF INDEX=NO OF LINES+1 THEN
H5: AL W3      , H4: 0 ;   GO TO FINISH;
W   HS. W2      H11. ;
    SN W2      0 ;   FOR I:=11 STEP -1 UNTIL 0 DO
    JL.        H3. ;   BEGIN
    SN W2      1 ;   IF I=0 THEN
    JL.        H6. ;   OUTCR ELSE
    SZ W2 000001 000000 ;
    SZ W2 111110 000000 ;   IF I=6 THEN
    JL.        H12. ;   OUTTAB ELSE
    AM          I15 ;
H6: AM          I14 ;
H7: AM          I13 ;
H13: AL W3      I6. ;
    JL. W2      H8. ;
    BZ. W2      H11. ;
    LS W2      -1 ;
    JL.        H5. ;
H12: SO W2      1 ;   IF -,BIT(I)OF:(LINE TABLE[INDEX]) THEN
    JL.        H7. ;   OUTSPACES ELSE
    SL W2 000001 000000 ;   BEGIN
    AM          I12 ;   K:=IF I>7 THEN 1 ELSE 2;
    AL W2      I11. ;
    RL W0      (X2) ;   OUTPUT(RESULT[POINTER[K]]);
    JL W3      (B24) ;
H   I6.      , 5 ;
W   RL W1      X2 ;   POINTER[K]:=POINTER[K]+2;
    AL W1      X1+2 ;
    RS W1      X2 ;
    JL.        H13. ;   END
                        ; END;
                        ; GO TO NEXT LINE;
H8: RS. W2      4 ;   PROCEDURE OUT (STRING);
    JL. W2      C14. ;   OUT STRING (STRING);
    0 ;

```

```

H9: JL. W3          C15. ; FINISH: TEST DEVICES(LOG SEMAPHORES);
H   A50 ,          A40 ; V(LOG SEMAPHORES);
W J5: 0
H   A50 ,          A32 ; EXIT PROGRAM;

```

```

W                                     ; STRING HEADING;
H I4: <<63>BALANCE:<0>>                ;
WHI1: <HHMM,<0>>                        ; STRING CLOCK STRING,
I2: <DD,MM.YYYY.<30>PERIOD:<0>>          ; DATE STRING,
WHI3: <HHMM,<63>>                        ; PERIOD STRING,
I5: <<63><10>>                          ; CR STRING,
I6: <DDDDD<0><10>>                      ; RESULT STRING,
I7: <<0><0><0><0><0><0><10>>              ; BLANK STRING,
I9: <<30><10>>                          ; TAB STRING;

```

```

W I10: 0 ; INTEGER ARRAY POINTER[1:2];
I11: 0
I16: 0 ; INTEGER LAST CALC;
H H11: 0

```

```

I12=I10-I11
I13=I7 -I6
I14=I5 -I7
I15=I9 -I5

```

```

; LINE TABLE:
; THE TABLE CONTAINS INFORMATION FOR THE FORMAT
; IN WHICH BALANCE RESULTS WILL BE PRINTED.
; EVERY BYTE DESCRIBES A PRINTED LINE OF 10 COLUMNS:
;      10 9 8 7 6 5 4 3 2
;      1 X X X X X 0 X X X X X
; WHERE X=1 MEANS THAT IN THE COLUMN A RESULT VALUE
; OTHERWISE BLANKS MUST BE OUTPUT.

```

H D1:

```

1 01111 0 10000
1 01111 0 10000
1 11111 0 00111
1 11111 0 00111
1 10111 0 10111
1 10111 0 10000
1 10000 0 10000
1 10000 0 10000
1 10000 0 00111
1 10000 0 00111
1 00111 0 10111
1 10000 0 00111
1 10000 0 10000

```

1	10000	0	00111
1	10000	0	00111
1	10000	0	10000
1	10000	0	10000
1	00111	0	00111
1	10111	0	10111
1	10111	0	10000
1	10000	0	10000
1	11111	0	10000
1	10000	0	10000
1	10000	0	10000
1	10111	0	10000
1	10000	0	10000
1	10000	0	10000
1	10000	0	10000
1	10000	0	11111
1	10000	0	11111
1	10000	0	01111

H14 = K-D1

E

; BALANCE PROGRAM

; LOG 1 PROGRAM:

; COMMENT IF THE BALANCE PROGRAM HAS BEEN EXECUTED
; IN BETWEEN THE LAST EXECUTION OF THIS PROGRAM, CR
; FOR SKIPPING THE LOG HEAD IS OUTPUT.

; THE PROGRAM LOGS DATA CORRESPONDING TO AD SCAN
; TABLE PART 1.

; IF AN AD SCAN TABLE IS SPECIFIED AS REMOVED OR
; CORRESPONDS TO A PASSIVE GROUP, BLANKS ARE
; OUTPUT.

; IF LIMITS ARE SPECIFIED FOR AN ENTRY, AND
; THE LIMITS ARE EXCEEDED ITS VALUE IS OUTPUT IN
; RED, OTHERWISE ITS VALUE IS OUTPUT IN BLACK.

; FOR ENTRY SPECIFIED AS NO LOGGING, NOTHING IS
; OUTPUT.

; CR ENTRY WILL CAUSE OUTPUT OF A CR.

; GROUP ENTRY WILL SET A BOOLEAN, CORRESPONDING
; TO THE GROUP STATE, ACTIVE OR PASSIVE.

; THE FIRST PART OF THE LOGGING IS TERMINATED WHEN
; AN END ENTRY IS SCANNED.

; THE PROGRAM LOGS THE INTEGRATED VALUES OF PULSES
; CORRESPONDING TO PULSE SCAN TABLE PART 1.

; CR ENTRY AND GROUP ENTRY IN THE PULSE SCAN TABLE
; HAS THE SAME EFFECT AS DESCRIBED ABOVE.

; LOG 1 PROGRAM TERMINATED WHEN AN END ENTRY
; IS FOUND IN THE PULSE SCAN TABLE.

; BY RUNMODE IS SPECIFIED:
; A: PRINTING ON LOG TWR 1
; B: PRINTING ON LOG TWR 2
; C: PRINTING ON PUNCH.
; NORMAL RUNMODE: A.

; NORMAL PERIODE: 1 HOUR
;

```

B      H7, J6, I5      ; BEGIN
W C4:  RL  W1          B37 ; LOG SEMAPHORES:=
      LS  W1          3   ; RUNMODE SHIFT 3 ^ 111 < 6;
      LA. W1          D5. ;
      RS. W1          J1. ;
      RS. W1          J6. ;
H      A50      ,      A38 ; P(LOG SEMAPHORES);
W J1:   0
H H1:  AL  W2      , F1: 8 ; INTEGER LOG 1 CR;
W      SL  W2      E5   ;
      JL.          H2. ; FOR I:= LOG 1 CR STEP 1 UNTIL
      AL  W2      X2+1 ; NO OF CR IN LOG 1 HEAD DO
      HS. W2      F1. ; BEGIN
      AL. W3      I4. ; LOG 1 CR:= I;
      JL. W2      H6. ; OUT(CR STRING);
      JL.          H1. ; END;
H2:    RL  W1          B38 ;
      JL  W3      (B19) ; CLOCK STRING:=
      0           ; CLOCK TO HOUR MIN (CLOCK);
      RS. W1          I2. ;
      AL. W3          I3. ;
      JL. W2          C12. ; TRANSFER DATE( DATE STRING);
      AL. W3          I5. ;
      JL. W2          H6. ; OUT (HEADING LOG 1);
      AL  W0          0   ; W0:= RELATIVE ENTRY AD SCAN TAB PART 1;
      AL  W2          F10 ; W2:= SCAN AD;
H3:    HS. W2          J2. ; INIT LOOP: CALL:= W2;
      RS. W0          J3. ; ENTRY:= W0;
F8:    ; STEP:
H H4:  JL. W2      , J2: 0 ; INTEGER CALL; GET LOG STRING (CALL, W3
W J3:   0           ; INTEGER ENTRY; ENTRY,
      J4:   0           ; INTEGER GROUP ACTIVE; GROUP,
      J5:   0           ; INTEGER INTGR NO; INTGR NO,
H      F3      , <<58><29>>; REL STRING ADDR, RED STRING,
F5:    <DDDD>          ; BLACK STRING.
      <<0><62><10><0>> ;
W      JL.          H5. ; GO TO TEST END);
      JL. W2      H6. ; OUT(W3);
      JL.          H4. ; GO TO STEP;
H5:    SO  W0          1   ; TEST END:
      JL.          H7. ; IF W0 = END 1 THEN GO TO FINIS;
      RL  W0          B33 ; W0 = FIRST ENTRY PULSE SCAN TAB PART 1;
      AL  W2          F11 ; W2 = SCAN INT;

```

```
      RL W1          B35 ; INTGR NO:=
      RS. W1         J5. ; FIRST ENTRY PULSE INTGR TAB PART 1;
      JL.            H3. ; GO TO INIT LOOP;

H6:   RS. W2          4   ; PROCEDURE OUT (STRING);
      JL. W2         C14. ; OUT STRING (STRING);
           0          ;
H7:   JL. W3         C15. ; FINIS: TEST DEVICES;
H     A50 ,         A40 ; V(LOG SEMAPHORES);
W J6: 0             ;
H     A50 ,         A32 ; EXIT PROGRAM;

H I5: <<63><62>LOG 1:<0><0>> ; STRING HEADING,
WHI2: <HHMM,<0>>          ; CLOCK STRING,
I3:   <DD,MM.YYYY.>        ; DATE STRING,
I4:   <<63><10>>          ; CR STRING;

E                                     ; END LOG 1 PROGRAM;
```

; LOG 2 PROGRAM:

; COMMENT THE PROGRAM LOGS VALUES CORRESPONDING
; TO AD SCAN TABLE PART 2 AND PULSE SCAN TABLE
; PART 2, AS DESCRIBED FOR LOG 1 PROGRAM.

; BY RUNMODE IS SPECIFIED;

; A: PRINTING ON LOG TWR 1

; B: PRINTING ON LOG TWR 2

; C: PRINTING ON PUNCH

; NORMAL RUNMODE: B

; NORMAL PERIOD : 1 HOUR

;

```

B      H7, J6, I5      ; BEGIN COMMENT SEE ALSO LOG 1 PROGRAM;

W C5:  RL  W1           B37 ; LOG SEMAPHORES:=
      ;                ; RUNMODE SHIFT 0 ^ 111 < 6;
      LA, W1           D5. ;
      RS, W1           J1. ;
      RS, W1           J6. ;
H      A50             A38 ;
W J1:   0
H H1:   AL  W2          , F2: 0 cf 26 ; INTEGER LOG 2 CR;
W      SL  W2           E6  ;
      JL.             H2.  ;
      AL  W2          X2+1 ;
      HS, W2          F2.  ;
      AL, W3           I4.  ;
      JL, W2           H6.  ;
      JL.             H1.  ;

H2:    RL  W1           B38 ;
      JL  W3           (B19) ;
      0
      RS, W1           I2.  ;
      AL, W3           I3.  ;
      JL, W2           C12. ;
      AL, W3           I5.  ;
      JL, W2           H6.  ;

      RL  W0           B31 ;
      AL  W2           F12 ;
H3:    HS, W2           J2.  ;
      RS, W0           J3.  ;

F9:
H H4:   JL, W2          , J2: 0 ;
W J3:   0
J4:   0
J5:   0
H      F4             , <<58><29>>;
F6:    <DDDD>
      <<0><62><10><0>>
W      JL.            H5.  ;
      JL, W2          H6.  ;
      JL.            H4.  ;

H5:    S0  W0           10  ;
      JL.            H7.  ;
      RL  W0           B15 ;
      AL  W2           F13 ;
      RL  W1           B35 ;

```

W0:= RELATIVE ENTRY AD SCAN TABLE PART 2;

W0:= FIRST ENTRY PULSE SCAN TAB PART 2;

```
AL W1      X1 E8      ; INTGR NO:=
RS. W1      J5.      ; FIRST ENTRY PULSE INTGR TAB PART 2;
JL.         H3.      ;

H6: RS. W2      4      ;
    JL. W2      C14.   ;
        0        ;
H7: JL. W3      C15.   ;
H   A50      ,    A40   ;
W J6:        0        ;
H   A50      ,    A32   ;

H I5: <<63><62>LOG 2:<0><0>> ; STRING HEADING,
WHI2: <HHMM,<0>>           ; CLOCK STRING,
I3:   <DD.MM.YYYY.>         ; DATE STRING,
I4:   <<63><10>>           ; CR STRING;

E                                     ; END LOG 2 PROGRAM
```

; ALARM SCAN PROGRAM:

; THE PROGRAM OUTPUTS ON THE STRIPPRINTER A
; TIME AND DATE INDICATION IN BLACK;

; DDDD-DDDD

; WHERE THE TWO FIRST DIGIT IS THE HOUR, THE
; TWO NEXT THE MINUTES, THEN TWO DIGIT FOR
; DAY AND TWO DIGIT FOR MONTH.

; THE PROGRAM SCANS THE AD SCAN TABLE.

; FOR EACH ENTRY ON WHICH TEST FOR HIGH AND/OR
; LOW LIMITS IS SPECIFIED, THE ENTRY VALUE IS
; INPUT, CONVERTED DO ENGINEERING UNIT AND
; TESTED AGAINST THE LIMITS.

; IF LIMITS ARE EXCEEDED OUTPUT ON THE STRIP-
; PRINTER IS MADE IN RED;

; DDD+DDDD L

; OR DDD+DDDD H

; WHERE THE FIRST NUMBER IS THE SCAN NUMBER,
; AND THE SECOND NUMBER IS THE SIGNED EN-
; GINEERING VALUE. L OR H INDICATE HIGH OR
; LOW LIMIT EXCEEDANCE.

; THE PROGRAM SCANS THE DIGITAL ALARM ENTRIES.

; IF AN ALARM CONDITION IS DETECTED OUT-
; PUT IS MADE ON THE STRIPPRINTER WITH RED:

; DDD-xxxxL

; WHERE THE NUMBER IS THE CORRESPONDING
; SCAN NUMBER.

; ONLY ENTRIES CORRESPONDING TO ACTIVE GROUPS
; ARE TESTED FOR ALARM CONDITIONS.

; THE PROGRAMS TERMINATES BY OUTPUT OF
; ALARM LAMPS. A LAMP IS SET FOR EACH
; GROUP IN WHICH AN ALARM CONDITION
; HAS BEEN FOUND.

; EXECUTION TIME NO ALARM: 4.5 SEC
; PER ALARM PRINTING: 1.0 SEC

; NORMAL PERIODE: 5 MIN.

```

B      H10, J8, I10      ; BEGIN
                        ; COMMENT ALARM SCAN PROGRAM;

H C6:      A50      ,      A38      ; P(STRIP SEMAPHORE);
W 19:      1<9      ;
W      RL W1      B38      ;
      JL W3      (B19)      ; CLOCK TO HOUR MIN(CLOCK);
      0      ;
      AL W0      12      ;
      LD W1      6      ;
      HS. W0      16.      ;
      BA. W1      19.      ;
      RS. W1      17.      ; CLOCK STRING:= CLOCK;
      RL W0      B41      ;
      RS. W0      18.      ; DATE STRING:= DAY + MONTH;
      AL. W3      16.      ;
      JL. W2      H8.      ; OUT(ALARM CHECK TIME);

      AL W3      0      ; ENTRY:= 0;
      RS W3      B43      ; COLL LAMPS:= 0;
      HS. W3      D6.      ; SCAN NO:= 0;

                        ; STEP 1:
H1:      JL W2      (B23)      ; NEXT ENTRY(ENTRY, TYPE);
      HS. W3      J2.      ;
      SZ W0 100000 000000      ; IF TYPE = -, AD ENTRY
      JL.      H3.      ; THEN GO TO TEST GROUP 1;
      BL. W2      D6.      ;
      AL W2      X2+1      ;
      HS. W2      D6.      ; SCAN NO:=SCAN NO+1;
H      AL W2      , J6: 0      ; BOOLEAN GROUP ACTIVE;
W      SZ W0 000000 100000      ;
      JL.      H1.      ; IF -, (TYPE=REMOVED
      SZ W0 001100 000000      ; ~ TYPE=HIGH ~ TYPE=LOW)
      SL W2      0      ; ~ -, GROUP ACTIVE
      JL.      H1.      ; THEN GO TO STEP 1;

      JL. W2      C9.      ; GET STRIP STRING (ENTRY, STRING,
      JL.      H2.      ; OK RETURN: GO TO TAKE STEP 1;
      JL. W2      H8.      ; OUT (STRING);
      RL W0      B43      ;
      LO. W0      J5.      ; COLL LAMPS:= COLL LAMPS ~ CUR LAMP;
      RS W0      B43      ;
H H2:      AL W3      , J2: 0      ; INTEGER NEW ENTRY;
                        ; TAKE STEP 1:
W      JL.      H1.      ; GO TO STEP 1;
H3:      AL. W2      H2.      ; TEST GROUP 1:
      SZ W0 001000 000000      ; IF TYPE = GROUP THEN
      JL.      H10.      ; BEGIN SET GROUP; GOTO TAKE STEP 1 END;
      SZ W0 000000 000010      ;

```



```

SZ W0 011000 000000 ; IF TYPE = END AD SCAN TAB FIRST PART
JL. H1. ; THEN GO TO STEP 1;
;
; READ DIGITAL ALARM DEVICES:
AL W1 A11 ; DEVICE:= FIRST DIGITAL ALARM;
AL W3 I1. ; FOR I:= 1 STEP 1 UNTIL 3 DO
JL W2 C13. ; BEGIN
RS W0 X3 ; DIGITAL VALUE(DEVICE, VALUE);
AL W1 X1+1<6-A24; DIG WORD[I]:= VALUE;
AL W3 X3+2 ; DIVICE:= DEVICE + 1;
SH W3 I3. ;
JL. C13. ; END;
;
; TEST DIGITAL ALARMS:
RL W3 B32 ; ENTRY:= ALARM SCAN TAB FIRST ENTRY;
AL W2 E7-2 ; SCAN NO:= FIRST ALARM NO - 1;
H5: BZ W0 X3 ; STEP 2:
AL W3 X3+1 ; TYPE := ALARM SCANTAB [ENTRY];
RS W3 J7. ; ENTRY:= ENTRY + 1;
SZ W0 100000 000000 ; IF TYPE = -, DIG ENTRY
JL. H7. ; THEN GO TO TEST GROUP 2;
AL W2 X2+1 ;
HS W2 D6. ; SCAN NO:= SCAN NO + 1;
LD W1 -4 ; WORD:=WORDPART(TYPE);
LS W1 -20 ; BIT:=BITPART(TYPE);
RL W1 X1+I1. ;
LS W1 (0) ;
BL W0 J6. ;
SH W0 -1 ; IF -, GROUP ACTIVE
SL W1 0 ; ~ DIG WORD [WORD, BIT] = 0
JL. H5. ; THEN GOTO STEP 2;
BL W0 D6. ;
JL W3 (B24) ; ALARM STRING:=
H W I4. , 3 ; CONV AN (SCAN NO);
RL W1 I4. ;
AL W1 X1 , I9:43 ; STRING <<0>->;
RS W1 I4. ;
AL W3 I4. ;
JL W2 H8. ; OUT(ALARM STRING);
RL W0 B43 ;
LO W0 J5. ; COLL LAMPS:= COLL LAMPS ~ CUR LAMP;
RS W0 B43 ;
H6: BL W2 D6. ; TAKE STEP 2:
RL W3 J7. ;
JL. H5. ; GO TO STEP 2;
;
H7: AL W2 H6. ; TEST GROUP 2:
SZ W0 001000 000000 ; IF TYPE = GROUP THEN
JL. H10. ; BEGIN SET GROUP; GO TO TAKE STEP 2 END;

```

```

SZ W0 011000 000000 ; IF TYPE = END THEN
JL. H6. ; GO TO TAKE STEP 2;
;
H RL W0 B43 ; END 2: W0:= COLL LAMPS;
A50 , A31 ; IO(DIG OUT, WRITE);
J8. , A15+A23 ;
W JL. H9. ; GOTO FINIS;
;
H8: RS. W2 4 ; PROCEDURE OUT (STRING);
JL. W2 C14. ; OUT STRING (STRING);
0 ;
H9: JL. W3 C15. ; FINIS: TEST DEVICES (STRIP SEMAPHORE);
H A50 , A40 ; V(STRIP SEMAPHORE)
W J10: 1<9 ;
H A50 , A32 ; EXIT PROGRAM;
;
W J5: 0 ; INTEGER CUR LAMP;
J7: 0 ; INTEGER ENTRY;
J8: 0 ; INTEGER STATUS DIG OUT
;
I1: 0 ; BOOLEAN ARRAY DIG WORD[1:4, 1:23];
0
0
I3: 0 ;
H I4: <DDD->, <<43><43><43><43><13><10>>;
; STRING DIG ALARM STRING;
I6: <<12>H> ; STRING ALARM CHECK TIME;
I7: <HMM-> ;
I8: <DDMM<12><10>> ;

```

```
; PROCEDURE SET GROUP;  
; REG: W0: TYPE / DESTROYED  
;       W1: DESTROYED  
;       W2: RETURN  
;       W3: DESTROYED
```

```
; CALL JL. W2 H10.
```

```
; COMMENT THE PROCEDURE UNPACKS A GROUP ENTRY,  
; SETS GROUP ACTIVE EQUAL TRUE OR FALSE AND  
; SELECT THE PROPER BIT FOR CURRENT LAMP;
```

```
W H10: LD  W1      -4      ; BEGIN  
      LS  W1      -20     ; GROUP NO:=GROUPNO PART(TYPE);  
      RL  W3      836     ;  
      LS  W3      X1-12   ;  
      HS. W3      J6.     ; GROUP ACTIVE:= GROUP [GROUP NO];  
      LD  W1      -5      ;  
      LS  W1      -19     ; LAMP NO:=LAMPNO PART(TYPE);  
      AL  W3      1       ;  
      LS  W3      X1      ; CURLAMP:= LAMP BIT [LAMP NO];  
      RS. W3      J5.     ;  
      JL  X2       ; END  
      ;  
E      ; END ALARM SCAN PROGRAM;
```

; TREND LOG:

;

; BY THE OPERATOR CONTROL PROGRAM THE OPERATOR CAN
; INPUT A SCAN NO, THE TREND SCAN NO, WHICH IS STORED
; IN THE ADDRESS CATALOG.

;

; THE PROGRAM FINDS THE ENTRY IN THE AD SCAN TABLE
; WHICH CORRESPONDS TO THE TREND SCAN NO.
; THE VALUE CORRESPONDING TO THE ENTRY IS MEASURED
; TESTED FOR LIMITS AND OUTPUT ON THE STRIPPRINTER.

;

; NORMAL PERIODE: DETERMINED BY OPERATOR.

;

; EXECUTION TIME: CA 1 SEC;

```

      4
B      H2, JR      ; BEGIN
      ; COMMENT TREND LOG;

H C7:      A50      ,      A38      ; P(STRIP SEMAPHORE);
W J2:      1<9      ;
      AL W3      0      ;
      AL W2      1      ; ENTRY:= 0
H1: HS. W2      D6,      ; SCAN NO:= 1
H2: JL W2      (B23)      ; FOR ENTRY:= NEXT ENTRY (ENTRY)
      SZ W0 100000 000000      ; WHILE TRUE DO
      JL.      H2,      ; BEGIN IF TYPE(ENTRY) = AD ENTRY THEN
      BL. W2      D6,      ; BEGIN SCAN NO:= SCAN NO + 1;
      AL W2      X2+1      ; IF SCAN NO = TREND SCAN NO
      SH W2      (B42)      ; THEN GO TO GET;
      JL.      H1.      ; END END;

      JL. W2      C9,      ; GET: GET STRIP STRING(ENTRY, STRING);
      AM      0      ;

      AL. W2      J2.      ;
      RS. W2      J1.      ;
      JL. W2      C14.      ; OUTSTRING(STRING);
J1:      0      ;
J2: JL. W3      C15.      ; TEST DEVICES (STRIP SEMAPHORE);
H      A50      ,      A40      ; V(STRIP SEMAPHORE);
W J2:      1<9      ;
H      A50      ,      A32      ; EXIT PROGRAM;

E      ; END TREND LOG;
L
; END S4 FIRST PART
N
T

```


; PROCEDURE GET LOG STRING (CALL);

; REG: W0: TYPE END OR DESTROYED
; W1: DESTROYED
; W2: LINK / DESTROYED
; W3: FREE / STRING ADDRESS

; CALL : W JL. W2 C8. OR C16.
; <ENTRY>
; <GROUP ACTIVE>
; <INTGR NO>
; H <REL STRING ADDR>
; <<58><29>> ; STRING RED STRING;
; <DDDD> ; STRING BLACK STRING;
; <ENDMARK> ; <0><10> OR <0><62>
; <<10><0>>
; W <END RETURN>: -
; <NORMAL RETURN>: -

; COMMENT THE PROCEDURE IS CALLED FROM THE PROGRAMS
; LOG 1 AND LOG 2. IT IS CALLED EITHER FOR EXAMINE
; AN AD SCAN TABLE ENTRY (C8.) OR A PULSE SCAN
; TABLE ENTRY (C16.).

; THE PROCEDURE RETURNS EITHER TO <NORMAL
; RETURN> WITH W3 CONTAINING THE ADDRESS OF
; A CHARACTER STRING, OR TO <END RETURN> WITH
; W0 CONTAINING THE END ENTRY FOUND.

; AD SCAN TABLE ENTRIES:

; AD ENTRY NOT MARKED FOR LOGGING: THE
; NEXT ENTRY IS EXAMINED.

; AD ENTRY MARKED REMOVED OR CORRESPONDING
; TO A PASSIVE GROUP: W3 POINTS ON BLANK STRING.

; OTHER AD ENTRIES: THE CORRESPONDING VALUE IS
; INPUT, AND CONVERTED TO ENGINEERING VALUE
; ON CHARACTER FORM. THE STRING IS STORED IN
; THE PARAMETER STRING: BLACK STRING. IF
; HIGH AND / OR LOW LIMIT ARE SPECIFIED FOR
; THE ENTRY TEST AGAINST THE LIMITS ARE PER-
; FORMED.

; IN CASE OF LIMIT EXCEEDANCE W3 POINTS
; ON RED STRING AND END MARK IS SET TO
; <0><62>, OTHERWISE W3 POINTS ON BLACK
; STRING AND END MARK IS SET TO <0><10>.

; GROUP ENTRY: DEPENDING ON THE CORRESPONDING
; GROUP NO THE PARAMETER <GROUP ACTIVE> IS

; SET FALSE OR TRUE. IF NO CR IS SPECIFIED
; THE NEXT ENTRY IS EXAMINED.

; CR ENTRY: W3 POINTS ON A CR STRING.

; PULSE SCAN TABLE ENTRIES:

; PULSE ENTRY: THE CONTENTS OF PULSE INTEGRATION
; TABLE ENTRY POINTED AT BY THE PARAMETER
; <INTGR NO> IS CONVERTED TO CHARACTERS,
; AND STORED IN THE PARAMETER <BLACK STRING>,
; <INTGR NO> IS COUNTED UP WITH 1. W3
; POINTS ON <BLACK STRING>.

; GROUP ENTRY: CR ENTRY: SEE ABOVE.

; WHEN RETURN THE PARAMETER <ENTRY> CONTAINS,
; THE RELATIVE ADDRESS OF THE EXAMINED ENTRY.

; THE PROCEDURE USES A SEMAPHORE;


```

B      H10, J4, I2      ; BEGIN

W C16: AM              I2      ; SWITCH STEP:= SCAN INT, SCAN AD;
C8:   AL. W1           H2,      ;
H      A50      ,      A38      ; P(GET LOG STRING);
W      1<5            ;
      RS. W1           J4,      ; STEP ADDR:= CALL;
      AL W2           X2+14      ;
      RS. W2           J3,      ; RETURN ADDR:= LINK + 14;
      HL W1           X2-8      ; STRING POINTER:=
      HS. W1           J2,      ; PARAM[REL STRING ADDR];
      JL.             (J4,)      ; GO TO STEP[STEP ADDR];

H1:   RL W3           X2-14      ; SCAN INT:
      BZ W0           X3      ; TYPE:=PULSE SCAN TAB[PARAM[ENTRY]];
      AL W3           X3+1      ; PARAM[ENTRY]:=
      RS W3           X2-14      ; PARAM [ENTRY] + 1;
      SZ W0 100000 000000      ; IF TYPE = PULSE ENTRY
      JL.             H6,      ; THEN GO TO TEST GROUP;
      RL W1           X2-10      ;
      RL W0           X1      ; W0:= PULSE INTGR TAB [PARAM[INTGR NO]];
      LS W0           -1      ; W0:=W0:2;
      AL W1           X1+2      ; PARAM[INTGR NO]:=
      RS W1           X2-10      ; PARAM [INTGR NO] + 2;
      RL W1           X2-12      ;
      SL W1           0      ; IF -, PARAM[GROUP ACTIVE]
      JL.             H9,      ; THEN GO TO OUT BLANK
      JL.             H3,      ; ELSE GO TO OUT BLACK;

H2:   RL W3           X2-14      ; SCAN AD: W3:= PARAM [ENTRY];
      JL W2           (B23)      ; NEXT ENTRY(W3, TYPE);
      RL. W2           J3,      ;
      RS W3           X2-14      ; PARAM[ENTRY]:= W3;
      HS. W3           J1,      ; CUR ENTRY:= W3;
      SZ W0 100000 000000      ; IF TYPE = AD ENTRY
      JL.             H6,      ; THEN GOTO TEST GROUP;
      RL W1           X2-12      ;
      SO W0 000010 000000      ; IF TYPE = LOGGING
      JL.             H2,      ; THEN GOTO SCAN AD;
      SO W0 000000 100000      ; IF TYPE = REMOVED ~
      SL W1           0      ; -, PARAM [GROUP ACTIVE]
      JL.             H9,      ; THEN GOTO OUT BLANK;
      JL W2           (B20)      ; VALUE (CUR ENTRY, W0);
H      AL W3           , J1: 0      ; INTEGER CUR ENTRY;
W      JL. W2           C10.      ; TEST LIMIT(CUR ENTRY, W0,
      0      ;
      JL.             H4,      ; HIGH: GOTO OUT RED;
      JL.             H4,      ; LOW: GOTO OUTRED);
H H3:  AL W3           , <<0><10>> ; OUT BLACK: END MARK:= <<0><10>>;

```

```

W      AL W2      -6      ; STRING ADDR:= BLACK STING;
      JL.          H5.      ; GOTO CONVERT;
H H4:  AL W3      , <<0><62>> ; OUT RED: END MARK:= <<0><62>>;
W      AL W2      -7      ; STRING ADDR:= RED STRING;
H5:    RL W1      J3.      ; CONVERT:
      HS W3      X1-4      ; CONV AN (W0,
      JL W3      (B24)      ; PARAM[BLACK STRING]);
F7:    ; USED FOR REL STRING ADDRESS DEFINITION;
H J2:    0      , 4      ; INTEGER STRING POINTER;
W      RL W3      J3.      ;
      WA W3      4      ;
      JL.          H10.     ; GO TO EXIT;

H6:    SZ W0 011000 000000 ; TEST GROUP: IF TYPE = END THEN
      JL.          H7.      ; BEGIN
      AL W2      X2-2      ; RETURN ADDR:= RETURN ADDR - 2;
      RS W2      J3.      ; GOTO EXIT;
      JL.          H10.     ; END;
H7:    LD W1      -4      ;
      SO W0 000000 100000 ; IF TYPE = GROUP
      JL.          H8.      ; THEN GOTO TEST CR;
      LS W1      -20      ; GROUP NO:= TYPE <9:12>;
      RL W3      B36      ; PARAM [GROUP ACTIVE]:=
      LS W3      X1      ; GROUP [GROUP NO] = ACTIVE;
      RS W3      X2-12     ;
H8:    SO W0 000001 000000 ; TEST CR: IF TYPE = CR
      JL.          (J4,)    ; THEN GOTO STEP [STEP ADDR];
      AM          -1      ; OUT CR: STRING ADDR:= CR STRING; GOTO EXIT;
H9:    AL W3      I1.      ; OUT BLANK: STRING ADDR:= BLANK STRING;

H H10:  A50      , A39      ; EXIT: VR(GET LOG STRING,
W      1<5      ; RETURN ADDR);
      J3: 0      ; INTEGER RETURN ADDR;
      J4: 0      ; INTEGER STEP ADDR;

H      <<63><10>>      ; STRING CR STRING;
I1:    <<0><0><0><0><0><10>> ; STRING BLANK STRING;

I2= H1 - H2      ;

E      ; END GET LOG STRING;

```

; PROCEDURE GET STRIP STRING

```

; REG:  W0:  DESTROYED
;        W1:  DESTROYED
;        W2:  LINK
;        W3:  ENTRY REL ADDRESS / STRIP STRING ADDRESS

```

```

; CALL:  JL, W2  C9,
;        <OK RETURN>;
;        <LIMIT RETURN>;

```

```

; COMMENT THE ENTRY VALUE IS INPUT AND CONVER-
;        TED TO ENGINEERING VALUE. THE ENGINEERING
;        VALUE IS CONVERTED TO CHARACTERS. IF HIGH AND/OR
;        LOW LIMIT TEST IS SPECIFIED FOR THE ENTRY
;        THE TEST IS MADE. DEPENDING ON THE RESULT
;        OF THE TEST THE CHARACTER STRING IS TERMINATED
;        BY THE CHARACTER FOR H-PRINT, L-PRINT OR
;        NORMAL-PRINT.
;        THE CURRENT SCAN NUMBER STORED IN D6, IS
;        CONVERTED TO CHARACTERS IN THE HEAD OF
;        THE STRING.
;        WHEN RETURN W3 CONTAINS THE ADDRESS OF THE
;        STRING.
;        RETURN IS MADE IN ACCORDANCE WITH THE
;        RESULT OF THE LIMIT TEST;

```

```

B      J3, I3      ; BEGIN
W C9:  HS. W3      J1,  ;
      RS. W2      J2,  ;
      JL W2      (B20) ; ENTRY VALUE (ENTRY,
      RS. W0      J3,  ; VALUE);
      SH W0      0    ;
      AC W0      (0)  ;
      JL W3      (B24) ; CONV AN (ABS (VALUE),
H      I2,      , 4    ; VALUE STRING, 4));
      AL W0      , D6: 0 ; INTEGER SCAN NO;
W      JL W3      (B24) ; CONV AN (SCAN NO,
H      I1,      , 3    ; SCAN NO STRING, 3);
W      RL. W0      J3,  ;
      RL. W1      I1,  ;
      SL W0      0    ; SIGN CHAR:= IF VALUE>0
      AM          1    ; THEN <+>
      AL W1      X1+42 ; ELSE <->;
      RS. W1      I1,  ; COMMENT <10> WOULD ACT AS ENDMARK;
H      AL W3      , J1: 0 ; INTEGER ENTRY;
W      JL. W2      C10. ;
      0            ; TEST LIMIT (ENTRY, VALUE,

```

```
AM          1<6      ;    HIGH: ENDPRINT:= H - PRINT;
AM          1<6      ;    LOW: ENDPRINT:= L - PRINT;
AL  W0      12<6+10  ;    OK: ENDPRINT:= NORMAL (PRINT);
HS. W0      I3.      ;    END STRING:= END PRINT CON <10>;

AL. W3      I1.      ;    W3:= STRIP STRING ADDRESS;
RL. W2      J2.      ;
SE  W0      12<6+10  ;    IF HIGH > LOW THEN
JL          X2+2      ;    GOTO LIMIT RETURN;
JL          X2        ;    GOTO OK RETURN;

J2:         0        ;    INTEGER RETURN;
J3:         0        ;    INTEGER VALUE;

;    STRING STRIP STRING,
H I1: <DDDS>        ;    SCAN NO STRING,
I2: <DDDD>        ;    VALUE STRING;
I3: <E<10>>        ;    END STRING;

E          ;    END GET STRIP STRING;
```

; PROCEDURE TEST LIMIT;

```

; REG:  W0:  VALUE
;        W1:  DESTROYED
;        W2:  LINK
;        W3:  ENTRY REL ADDR / DESTROYED

```

```

; CALL:      JL. W2      C10.
;           <WORKING LOCATION>
;           <HIGH LIMIT RETURN>;
;           <LOW LIMIT RETURN>;
;           <OK RETURN>;

```

```

; COMMENT THE TYPE OF THE ENTRY IS EXAMINED, IF
; HIGH AND/OR LOW LIMITS ARE SPECIFIED THE VALUE
; IS TESTED AGAINST THE LIMITS. RETURN IS MADE
; IN ACCORDANCE WITH THE RESULT OF THE TEST;

```

```

B      H1                      ; BEGIN

W C10: WA  W3          B30    ;
      RS  W0          X2      ;
      BZ  W1          X3      ; TYPE:= SCAN TAB [ENTRY];
      SO  W1 001000 000000 ; TEST HIGH:
      JL.          H1.      ; IF TYPE = HIGH
      BL  W0          X3+1    ; ^ W0 > SCAN TAB[HIGH LIMIT]
      RX  W0          X2      ;
      SL  W0          (X2)    ;
      JL          X2+2      ; THEN HIGH LIMIT RETURN;
      AL  W3          X3+1    ;
      RS  W0          X2      ;
H1:    SO  W1 000100 000000 ; TEST LOW:
      JL          X2+6      ; IF TYPE = LOW
      BL  W0          X3+1    ; ^ W0 < SCAN TAB [LOW LIMIT]
      RX  W0          X2      ;
      SH  W0          (X2)    ;
      JL          X2+4      ; THEN LOW LIMIT RETURN;
      JL          X2+6      ; OK RETURN;

E                      ; END;

```

; PROCEDURE TRANSFER DATE

;

; REG: W0: DESTROYED

; W1: SAVED

; W2: RETURN

; W3: STRING ADDRESS

;

; CALL: JL, W2 C12.

;

; COMMENT THE STRING ADDRESS MUST BE AN ODD NUMBER,

; FORMAT OF STRING:

; DD.MM.YYYY;

B I1 ;

W C12: BZ W0 B41 ;

HS W0 X3 ;

BZ W0 1+B41 ;

LS W0 6 ;

WA, W0 I1. ;

RS W0 X3+1 ;

RL W0 2+B41 ;

RS W0 X3+3 ;

JL X2 ;

I1: <.<0><0>.> ;

E

; PROCEDURE DIGITAL VALUE;

```

; REG:  W0:  - / VALUE
;        W1:  DEVICE NO <6 / DEVICE NO <6 + A24
;        W2:  LINK
;        W3:  SAVED

```

; CALL: JL, W2 C13.

```

; COMMENT A DIGITAL REGISTER IS INPUT. ITS VALUE
;        IS GIVEN IN W0 WHEN RETURN;

```

```

C      A56<4,J1,I2          ; IF NO SIMULATION THEN
                                ; BEGIN

H C13:      A50      ,      A38 ; P(DIGITAL)
W                                ;
      HS. W1          I2. ;
      AL W1          X1+A24 ;
      HS. W1          I1. ;
H      A50      ,      A31 ; IO(DEVICE, NO OPR, VALUE);
      J1.      , I1: 0 ;
      A50      ,      A31 ; IO(DEVICE, SENSE)
      0      , I2: 0 ;
W      RL. W0          J1. ; W0:= VALUE;
H      A50      ,      A40 ; V(DIGITAL);
W                                ;
      JL          1<10 ;
      JL          X2 ; GOTO LINK;

      J1:      0 ; INTEGER VALUE;

E                                ; END

C      [A56>19^1]-1 ; ELSE
W C13:  LS W1          -5 ; VALUE:=SIMULATION TABLE
      AM          (B18) ; 2*DEVICE NO-32];
      RL W0          X1-32 ;
      LS W1          5 ;
      AL W1          X1+A24 ;
      JL          X2 ;

E

```

; LOG DEVICE PROCEDURES:

; PRINTING CAN BE PERFORMED ON FOLLOWING DEVICES:

```
;          LOG TWR 1
;          LOG TWR 2
;          PUNCH
;          STRIPPRINTER
```

; THERE IS ONE PROCEDURE FOR PRINTING OCCURED
; ERRORS, CALLED: TEST DEVICES.

; WHEN A PROGRAM USE ONE OR SOME DEVICES, THE
; PROGRAM MUST RESERVE THE DEVICES BY A P CALL
; WITH THE DEVICE SEMAPHORES,

; WHEN A PROGRAM FINISHES USE OF DEVICES, IT MUST
; MAKE A CALL OF TEST DEVICES AND A V CALL
; WHICH RELEASES THE DEVICES,

; THE SEMAPHORES ARE GIVEN AS PARAMETER TO THE
; PROCEDURE CALLS, OUT STRING WILL OUTPUT ON THE
; DEVICES WHICH CORRESPONDS TO THE SEMAPHORES,
; AND TEST DEVICE WILL TEST THE CORRESPONDING
; DEVICES.

; PROGRAMS WHICH OUTPUT ON STRIPPRINTER, ALARM
; SCAN, AND TRENDLOG WILL NOT OUTPUT ON OTHER
; DEVICES, DUE TO THE DIFFERENT CHARACTER SET.

; FOR THE PROGRAMS: BALANCE, LOG 1, AND LOG 2
; THE OPERATOR CAN DETERMINE WHICH OF THE
; DEVICES, LOG TWR 1, LOG TWR 2, AND / OR PUNCH, SHALL
; BE USED FOR OUTPUT.

; THE INFORMATION IS STORED IN A B TABLE
; WORD, RUNMODE, IN SUCH A WAY THAT THE
; RELEVANT INFORMATIONS EASILY CAN BE EXTRACTED
; AND USED DIRECTLY AS SEMAPHORE PARAMETERS,

; IN A PROGRAM THE FOLLOWING CODE WILL
; OCCUR:

```
;      RL  W  <RUNMODE>                      ;
;      LS  W  <PROGRAM DEPENDING NUMBER>      ;
;      LA  W  <ADDRESS OF MASK>                ;      W:= SEMAPHORES;
;      RS. W      J1.                          ;
;      RS. W      J2.                          ;
;      A50      ,  A38                        ;      P(SEMAPHORES)
; J1:      0                                ;      INTEGER SEMAPHORES;
```


; THIS WILL UNPACK RUNMODE AND SET THE
; SEMAPHORES FOR THE P AND V CALL, AND EXECUTE
; A P CALL,
; FOR CALLING OUTSTRING AND TEST DEVICES,
; AND THE V FOLLOWING CODE OCCUR:

; RS. W2	4	; PROCEDURE OUT (STRING)
; H1: JL. W2	<OUTSTRING>	; OUTSTRING(STRING);
; 0		;
; JL. W3	<TEST DEVICES>	; TEST DEVICE;
; A50 , A40		; V(SEMAPHORES);
; J2: 0		; INTEGER SEMAPHORES;

; DUE TO THE FIXED FORMAT THE PROCEDURES OUTSTRING,
; AND TEST DEVICES CAN USE SEMAPHORES STORED
; FOR THE V CALL.

; INSIDE THE PROGRAM THE LOCAL PROCEDURE
; IS CALLED:
; W3 = ADDRESS OF STRING ;
; JL. W2 H1.

; PROCEDURE OUTSTRING (STRING ADDR);

```

; REG:  W0:  DESTROYED
;        W1:  DESTROYED
;        W2:  LINK
;        W3:  STRING ADDRESS / DESTROYED

```

```

; CALL:      JL, W2      C14,
;            <RETURN ADDRESS>
;            <NOT USED>
;            <NOT USED>
;            <SEMAPHORES>

```

```

; COMMENT A CHARACTER STRING IS PRINTED ON THE
;         DEVICES SPECIFIED BY THE SEMAPHORES. THE CHARACTER
;         STRING SHALL BE TERMINATED WITH THE CHARACTER
;         <10>. THE OUTPUT OPERATIONS CONSIST ONLY OF
;         A WRITE. THE STATUS FOR A DEVICE IS
;         AFTER EACH WRITE OPERATION LOGICALLY ADDED
;         TO PREVIOUSLY COLLECTED STATUS.

```

B H11, J8

```

W C14:      ; BEGIN
H1:  BZ  W0      X3      ; UNPACK: FIRST:= TRUE;
      LD  W1      -6      ; W0:= LEFT CHAR; W1:= RIGHT CHAR;
      AL  W3      X3+1    ; STRING ADDR:= STRING ADDR + 1;
H2:  SN  W0      10      ; TEST CHAR: IF W0 = 10
      JL      (X2)      ; THEN GOTO WORD [LINK];
      SN  W0      63      ; IF W0 = 63
      AL  W0      64      ; THEN W0:= 64;
      RX  W1      X2+6    ; SEMAPHORE:= WORD[LINK + 6];

      SO  W1      1<6    ;
      JL.      H3,      ; IF SEMAPHORE TWR 1
H      A50      ,      A31 ; THEN IO(W0, TWR, WRITE);
      J1.      , A3+A23    ;

W H3:  SO  W1      1<7    ;
      JL.      H4,      ; IF SEMAPHORE TWR 2
H      A50      ,      A31 ; THEN IO(W0, TWR 2, WRITE);
      J3.      , A4+A23    ;

W H4:  SO  W1      1<8    ;
      JL.      H5,      ; IF SEMAPHORE PUNCH
H      A50      ,      A31 ; THEN IO(W0, PUNCH, WRITE);
      J5.      , A1+A23    ;

W H5:  SO  W1      1<9    ;
      JL.      H6,      ; IF SEMAPHORE STRIPPRINTER

```

```

C      A56<12      ; THEN BEGIN
H      A50      ,      A31      ; IF STRIPPRINTER NOT SIMULATED THEN
      J7.      ,      A5+A23      ; IO(W0,STRIPPR,WRITE)

E      [A56>11^1]-1,H0      ; ELSE
C      SN W0      42      ; BEGIN TRANSLATE CHARACTERS:
W      AL W0      32      ; <MINUS> -> -
      SN W0      43      ;
      AL W0      27      ; <PLUS> -> ,
      SN W0      12      ;
      AL W0      37      ; <NORMAL CR> -> N
      SN W0      13      ;
      AL W0      35      ; <LOW CR> -> L
      SN W0      14      ;
      AL W0      56      ; <HIGH CR> -> H
      SH W0      34      ; IF W0=L^W0=H^W0=N THEN
      JL.      H0.      ; BEGIN
H      A50      ,      A31      ; IO(W0,LOG 1,WRITE);
W      J7.      ,      A3+A23      ; W0:=-<CR>
      AL W0      64      ; END;
H H0:      A50      ,      A31      ; IO(W0,LOG 1,WRITE)
      J7.      ,      A3+A23      ; END
E      ; END;

W H6:      RL. W0      J1.      ; COLL TWR 1:= COLL TWR 1
      LO. W0      J2.      ; ~ STATUS TWR 1;
      RS. W0      J2.      ;
      RL. W0      J3.      ; COLL TWR 2:= COLL TWR 2
      LO. W0      J4.      ; ~ STATUS TWR 2;
      RS. W0      J4.      ;
      RL. W0      J5.      ; COLL PUNCH:= COLL PUNCH
      LO. W0      J6.      ; ~ STATUS PUNCH;
      RS. W0      J6.      ;
      RL. W0      J7.      ; COLL STRIPPR:= COLL STRIPPR
      LO. W0      J8.      ; ~ STATUS STRIPPR
      RS. W0      J8.      ;
      AL W0      0      ;

      RX W1      X2+6      ; SEMAPHORE CROSS W1;
      AC W3      X3      ; FIRST:= -, FIRST;
      SL W3      0      ; IF -, FIRST THEN GOTO UNPACK;
      JL.      H1.      ;
      LD W1      6      ; W0:= RIGHT CHART;
      JL.      H2.      ; GOTO TEST CHAR;
      ; END PROCEDURE OUT STRING;

```

; PROCEDURE TEST DEVICES;

; REG: W0: DESTROYED
; W1: DESTROYED
; W2: DESTROYED
; W3: LINK

; CALL: <ERROR RETURN>: JL. W3 C15.
; <OK RETURN>: <INSTRUCTION>
; <SEMAPHORES>

; COMMENT THE PROCEDURE SENSES THE DEVICES WHICH
; IS SPECIFIED BY THE SEMAPHORES, AND TESTS
; WHETHER THE COLLECTED STATUS IS ZERO.
; IF ERRORS IS FOUND, ERROR MESSAGE IS GIVEN
; ON THE CONTROL TYPEWRITER, THE COLLECTED
; STATUS WITH THE ERROR IS RESET TO ZERO,
; AND THE PROCEDURE IS CALLED ONCE MORE.

```

; BEGIN
C15: RL W1 X3+2 ;
      SO W1 1<6 ;
      JL. H7. ; IF SEMAPHORE TWR 1
H      A50 , A31 ; THEN IO(TWR 1, SENSE);
      0 , A3 ;
W      JL. W2 H11. ; TEST (STATUS TWR 1, COLL TWR 1);
      J1: 0 ; INTEGER STATUS TWR 1;
H      J2: 0 , A3 ; INTEGER COLL TWR 1;

W H7: SO W1 1<7 ;
      JL. H8. ; IF SEMAPHORE TWR 2
H      A50 , A31 ; THEN IO(TWR 2, SENSE);
      0 , A4 ;
W      JL. W2 H11. ; TEST (STATUS TWR 2, COLL TWR 2);
      J3: 0 ; INTEGER STATUS TWR 2;
H      J4: 0 , A4 ; INTEGER COLL TWR 2;

W H8: SO W1 1<8 ;
      JL. H9. ; IF SEMAPHORE PUNCH
H      A50 , A31 ; THEN IO(PUNCH, SENSE);
      0 , A1 ;
W      JL. W2 H11. ; TEST (STATUS PUNCH, COLL PUNCH);
      J5: 0 ; INTEGER STATUS PUNCH;
H      J6: 0 , A1 ; INTEGER COLL PUNCH;

W H9: SO W1 1<9 ;
      JL. H10. ; IF SEMAPHORE STRIPPR
H      A50 , A31 ; THEN IO(STRIPPR, SENSE);
      0 , A5 ;
W      JL. W2 H11. ; TEST (STATUS STRIPPR, COLL STRIPPR);
      J7: 0 ; INTEGER STATUS STRIPPR;
H      J8: 0 , A5 ; INTEGER COLL STRIPPR;
W H10: JL X3 ; GOTO LINK;

H11: BZ W0 X2+2 ; PROCEDURE TEST (STATUS , COLL , DEVICE NO);
      LO W0 X2 ;
      SN W0 0 ;
      JL X2+4 ; IF STATUS=0 & COLL=0 THEN
      BZ W1 X2+3 ; BEGIN COLL:= 0;
      RS W1 X2+2 ; W1:= DEVICE NO;
      AL W0 0 ; W0:= 0;
      RS W0 X2 ; STATUS:=0;
      AL W2 X3-2 ; DEVICE ERROR MESSAGE;
      JL W3 (B21) ; END;
      JL X2 ;

E ; END PROCEDURE TEST DEVICE;

```



```
; C1 PROGRAM: PULSE SCAN
; C2 - : FLOW SCAN
; C3 - : BALANCE
; C4 - : LOG 1 SCAN
; C5 - : LOG 2 SCAN
; C6 - : ALARM SCAN
; C7 - : TREND LOG
;
; C8 PROCEDURE: GET LOG STRING
; C9 - : GET STRIP STRING
; C10 - : TEST LIMIT
; C12 - : TRANSFER DATE
; C13 - : DIGITAL VALUE
; C14 - : OUT STRING
; C15 - : TEST DEVICES
; C16 PROGRAM : BALANCE RESET PHASE
```

```
; D1 TABLE : LINE TABLE
W D2: 86400 : DAY IN SECOND
W D5: 111<6 : MASK SEMAPHORE
; D6 : SCAN NO
```

```
; POINTS AND DEFINITIONS ACROSS BLOCKS
```

```
; F1 VARIABLE: LOG 1 CR
; F2 - : LOG 2 CR
```

```
F3 = F5 - F7 ; REL STRING ADDR LOG 1
F4 = F6 - F7 ; - - - LOG 2
; F5 POINT STRING ADDR LOG 1
; F6 - STRING ADDR LOG 2
; F7 - USE STRING ADDRESS, GET LOG STRING;
```

```
; F8 POINT LOG 1 FOR CALL OF GET LOG STRING
; F9 - LOG 2 - - - - -
```

```
F10 = C8 - F8; LOG 1: GET LOG STRING (SCAN AD)
F11 = C16- F8; - - - (SCAN INT)
```

```
F12 = C8 - F9; LOG 2: - - - (SCAN AD)
F13 = C16- F9; - - - (SCAN INT)
```

```

B      E5                      ; TAIL SEGMENT 4:
W

E4:  AL  W2          0      ;  INITIALIZE(
E5:  RL  W0      X2+E1.    ;    PULSE SCAN,
      RS          X2+A46+2  ;    FLOW SCAN,
      RL  W1      X2+E2.    ;    BALANCE,
E3:  AL  W0      X1+E3.    ;    LOG 1,
      RS  W0      X2+A47+2  ;    LOG 2,
      RL  W0      X2+E0.    ;    ALARM SCAN,
      RS  W0      X2+A49+2  ;    TREND LOG,
      AL  W0          0      ;
      RS  W0      X2+A48+2  ;
      AL  W2      X2+2      ;
      SE  W2          14     ;
      JL.          E5.      ;

      AL  W0          C17.   ;    BALANCE RESET,
      RS  W0          B47    ;
      AL  W0          E4.    ;    FIRST FREE);
      AL  W1          0      ;
      AL  W2      A54>19^1  ;
      JL  W3      (B39)     ;    CALL LOADER;

E1:  <PULS>          ;    USER NO 1
      <FLOW>          ;           2
      <BALA>          ;           3
      <LOG1>          ;           4
      <LOG2>          ;           5
      <ALAR>          ;           6
      <TREN>          ;           7

E2:  C1-E3           ;    PULSE
      C2-E3           ;    FLOW
      C3-E3           ;    BALANCE
      C4-E3           ;    LOG 1
      C5-E3           ;    LOG 2
      C6-E3           ;    ALARM
      C7-E3           ;    TREND LOG

E0:                      ;    NORMAL PERIOD:
      1                ;    1 SEC
      300              ;    5 MIN
      28800            ;    8 HOURS
      3600             ;    1 -
      3600             ;    1 -
      300              ;    5 MIN
      1                ;    1 SEC

      JL.          E4.    ;

```


E
L
; IDLIST S4
N
T
E
E
L
; END S4
N
T

C [A55>18^1] - 1 ;
S C5, D1, G20, I15 ;
W K=0 ; SEGMENT 5:

; THIS SEGMENT CONTAINS THE PROCEDURE FOR
; INPUTING ANY A/D MEASUREMENT, AND HANDLING
; THE SCAN TABLE.

; PROCEDURE VALUE SUPPLIES THE RESULT OF A
; MEASUREMENT. THE PHYSICAL QUANTITY TO BE
; MEASURED IS DESCRIBED IN THE SCAN TABLE.
; THIS DESCRIPTION CONSIST OF (AMONG OTHERS, SEE
; SEGMENT 10) THE A/D TERMINAL ADDRESS, THE
; NUMBER OF THE EQUATION TO BE USED FOR
; CONVERSION TO ENGINEERING VALUE, AND
; SEVERAL PARAMETERS FOR THE EQUATION. AMONG
; THESE PARAMETERS MAY BE SOME REFERENCES
; TO ANOTHER QUANTITIES, WHICH CAUSE THE
; CALLING OF THE PROCEDURE RECURSIVLY. HOWEVER,
; AS THE SYSTEM DOES NOT USE ANY STACK, THIS
; POSSIBILITY IS LIMITED.

```

; INTEGER PROCEDURE VALUE(ENTRY);
; REG      W0   FREE / VALUE
;          W1   FREE / DESTROYED
;          W2   RETURN/ DESTROYED
;          W3   ENTRY / DESTROYED
; CALL     JL W2 (B20) OR JL W2 C4. INTERNALLY
; TEST OUTPUT: EQ <EQUATION NO>
;          AD OR AS <A/D INPUT VALUE>
;          RE <RESULT>
; EQ AND AD, AS CAN APPEAR SEVERAL TIMES IN
; THE CASE OF RECURSIVE CALLING

; SEMAPHORE: 1<3

```

```

B H2
H C0:      A50      ,      A38      ; BEGIN
W          1<3      ; P(3);

      RS. W2      H0.      ;
      AL. W2      H2.      ;
C4:      WA W3      B30      ; ENTRY:= ENTRY + TABLE BASE+1;
      BZ W0      X3      ; K:= STORE[ENTRY-1];
      AL W3      X3+1      ;
      LD W1      -5      ;
      LS W1      -19      ; M:= EQREF PART (K);
      SZ W0      8      ; IF L BIT (K) THEN
      AL W3      X3+1      ; ENTRY:= ENTRY + 1;
      SZ W0      16      ; IF H BIT (K) THEN
      AL W3      X3+1      ; ENTRY:= ENTRY + 1;
C W      [A58>18^1] - 1      ; IF TESTOUTPUT THEN
H E      AL W0      X1      ;
      A51      ,      <EQ>      ; TESTOUT(<EQ>, M);
      ;
      RZ. W1      X1+D1.      ;
      LA. W1      G12.      ;
      JL.      X1+D0.      ; CALL EQUATION (EQTABLE[M]);
H2:      ; VALUE:= E;
C W      [A58>18^1] - 1      ; IF TESTOUTPUT THEN
H E      A51      ,      <RE>      ; TESTOUT(<RE>, VALUE);
      ;
      RL. W2      H0.      ;
H W      A50      ,      A40      ;
H W      1<3      ; V(3);
W      A50      ,      A37      ; PASS CONTROL;
      JL      X2      ; END;

```

H0: 0 ;

E

```
; EQTABLE[0:15]
; FORMAT:
; <NO OF PARAMETERS + 1><9 + <ENTRY TO THE CODE> - D0
; THE EQUATIONS ARE ARRANGED INTO THREE GROUPS
; DEPENDING ON THE NUMBER OF CONSTANTS THEY USE:
; 1ST GROUP: C1 , C2
; 2ND - C1
; 3RD - NO CONSTANTS ARE USED
; WHEN CHANGING THE TABLE, ATTENTION SHOULD BE DRAWN
; TO SEGMENT 3 ( ITEM NO. CHECKING ) AND SEGMENT 11
; ( DEFINITION OF E NAMES )
```

W G12: 1<9-1 ; CONSTANT

H D1: I0 ; EQ 00

I1 ; EQ 2

I2 ; EQ 3

I3 ; EQ 5

I4 ; EQ 0

I5 ; EQ 1

I6 ; EQ 4

I7 ; EQ 10

I8 ; EQ 15

I9 ; EQ 16

I10 ; EQ 17

I11 ; EQ 6

I12 ; EQ 11

I13 ; EQ 12

I14 ; EQ 13

I15 ; EQ 14

0 ; EQ 18

G14: 857 ; BYTE CONSTANT

```
;
;   PROCEDURE NEXT ENTRY ANALYSES A GIVEN
;   RECORD IN THE SCAN TABLE, AND GIVES A
;   REFERENCE TO THE NEXT ONE.
```

```
; PROCEDURE NEXT ENTRY(ENTRY);
; REG   W0   FREE / SCAN TABLE[NEXT ENTRY]
;       W1   FREE / NO OF CONSTANTS + 1
;       W2   LINK
;       W3   ENTRY / NEXT ENTRY
; CALL   JL W2 (B23)
```

```
B H0
W C3: AM           (B30) ; BEGIN
      BL W0       X3     ; K:= SCAN TABLE[ENTRY];
      AL W3       X3+1   ; ENTRY:= ENTRY + 1;
      SH W0       -1     ; IF K < 0 THEN
      JL.         H0.    ; GO TO END;
      LD W1       -5     ;
      LS W1       -19    ; ENTRY:= ENTRY +
      BZ. W1      X1+D1.  ; CONSTPART(EQTABLE[
      LS W1       -9     ; EQREFPART(K)]) +
      SZ W0       8      ; (IF -, L BIT(K) THEN 0
      AL W3       X3+1   ; ELSE 1) +
      SZ W0       16     ; (IF -, H BIT(K) THEN 0
      AL W3       X3+1   ; ELSE 1);
      WA W3       2      ;
      H0: AM      (B30) ; END:
      BL W0       X3     ; NEXT ENTRY:= ENTRY
      D0: JL      X2     ;
E                                     ; END:
                                     ; EQ 18:
                                     ; RETURN;
```

```

B      H8, J3
W      ; EQ 1, PARAM: C1, P, T
      ; E = SQRT(A*(P+10)*2^10/(T+273)*2^10/Z)*C1*2^(-11)
      ; Z = (857+68*T/50)+(857+68*T/50-1000)*((P+10)-600)/600

      I5=4<9+K-D0 ;
      RS. W2      H0. ; EQ1:
      RS. W3      H1. ;
      BZ W3      X3+3 ;
      JL. W2      C4. ; T:= VALUE(STORE[ENTRY + 3]);
      RS. W0      H2. ;
      WM. W0      G1. ;
      WD. W0      G2. ;
      SL W3      25 ;
      BS. W0      G13. ; K:= ENTIER(T*68/50 + 0.5);
      BA. W0      G14. ;
      RS. W0      H3. ; M3:= K + 857;
      WS. W0      G9. ;
      RS. W0      H4. ; M4:= K + 857 - 1000;
      AM.      (H1.) ;
      BZ W3      2 ;
      JL. W2      C4. ; P:= VALUE(STORE[ENTRY + 2]);
      RL. W2      H1. ;
      BZ W3      X2+1 ; C1:= STORE[ENTRY + 1];
      RS. W3      H1. ;
      WA. W0      G4. ;
      RS. W0      H5. ; M5:= P + 10;
      WS. W0      G6. ;
      WM. W0      H4. ; K:= ((P + 10) - 600) * M4;
      WD. W0      G6. ;
      SL W3      300 ;
      RS. W0      G13. ; K:= ENTIER(K/600 + 0.5);
      WA. W0      H3. ;
      AL W3      (0) ; Z:= K + M3;
      BZ W0      X2 ;
      JL. W2      C2. ; A:= AD(STORE[ENTRY]);
      WM. W0      H5. ;
      RL. W1      H2. ;
      AL W1      X1+273 ;
      AD W0      10 ;
      WD W0      2 ; K:= A * M5 * 2 ^ 10 : (T + 273);
      AL W3      0 ;
      AD W0      10 ;
      WD W0      4 ; K:= K * 2 ^ 10 : Z;
      JL. W3      C1. ;
      WM. W0      H1. ;
      AD W0      -11 ; E:= SQRT(K) * C1 * 2 ^ (-11);
      JL.      (H0.) ; RETURN;

```

```

G1:          68      ;   CONSTANTS
G2:          50      ;
G6:          600     ;

H0:          0       ;   WORKING LOCATIONS FOR
H1:          0       ;   EQ 1, 2, 3, 4, 5,10,15,16,17
H2:          0       ;
H3:          0       ;
H4:          0       ;
H5:          0       ;

```

```

; EQ 5 , PARAM: C1, C2, P
; E = A*(C1*(P+C2)*2^(-11))*2^(-10)

```

```

I3=4<9+K-D0      ; EQ5:
AM                24      ;   EQ5:= TRUE; GO TO TAKE P;

```

```

; EQ 3 , PARAM: C1, C2, P
; E = C1*SQRT(A*(P+C2))*2^(-10)

```

```

I2=4<9+K-D0      ; EQ3:
AL  W0            24      ;   EQ3:= TRUE; EQ5:= FALSE;
AM                (0)     ;   GO TO TAKE P;

```

```

; EQ 2 , PARAM: C1, C2, P, T
; E = C1*SQRT(A*(P+C2)/(273+T)*2^8)*2^(-10)

```

```

I1=5<9+K-D0      ; EQ2:
LD  W1            -48     ;   EQ3:= EQ5:= FALSE;
HS. W0            H6.     ;
HS. W1            H7.     ;   TAKE P:
RS. W2            H0.     ;
RZ  W1            X3+3     ;   K:= STORE[
AM                (B30)    ;   STORE[ENTRY+3] + TABLE BASE];
BZ  W0            X1       ;
JL. W2            C2.     ;   P:= AD(K);
BA  W0            X2+2     ;
RS. W0            H1.     ;   M1:= P + C2;
AL  W3            X2       ;
BZ  W0            X2+1     ;
RS. W0            H2.     ;   M2:= C1;
RZ  W0            X2       ;
JL. W2            C2.     ;   A:= AD(STORE[ENTRY]);
SE  W3            , H6: 0   ;   IF EQ 5 THEN
JL. W0            J0.     ;   GO TO FIVE;
WM. W0            H1.     ;   K:= A * M1;
SE  W3            , H7: 0   ;   IF EQ 3 THEN
JL. W0            J1.     ;   GO TO SQRT;
RS. W0            H1.     ;   M1:= K;

```

H
W
H
W


```

      BZ  W3      X2+4      ;
      JL  W2      C4.      ;   T:= VALUE(STORE[ENTRY + 4]);
      AL  W1      (0)      ;
      AL  W1      X1+273    ;
      RL  W0      H1.      ;
      AL  W3      0        ;
      WD  W0      2        ;
      AS  W0      8        ;   K:= M1 / (T + 273) * 2 ^ 8;
J1:   JL  W3      C1.      ;   SQRT:
      WM  W0      H2.      ;   K:= SQRT (K) * M2;
J2:   AD  W0      -10      ;   SHIFT:
      JL  (H0.)    ;   E:= K * 2 ^ (-10);
      JL  (H0.)    ;   RETURN;
      JL  (H0.)    ;   FIVE:
J0:   RL  W2      H1.      ;
      WM  W2      H2.      ;
      AD  W2      -11      ;
      WM  W0      4        ;   K:= M1 * M2 * 2 ^ (-11) * A;
      JL  J2.      ;   GO TO SHIFT;

```

```

; EQ 4 , PARAM: C1
; E = C1 * SQRT(A * 2 ^ 12) * 2 ^ (-11)

```

```

      I6=2<9+K-D0      ; EQ4:
J3:   RS  W2      H0.      ;
      BZ  W0      X3      ;
      JL  W2      C2.      ;   K:= AD(STORE[ENTRY]);
      AS  W0      12      ;
      JL  W3      C1.      ;   K:= SQRT(K * 2 ^ 12);
      BZ  W1      X2+1    ;
      WM  W0      2        ;
      AD  W0      -11      ;   E:= C1 * K * 2 ^ (-11);
      JL  (H0.)    ;   RETURN;

```

```

; EQ 15 , PARAM: C1, T1, T2
; F = F * (T2 - T1) / 100

```

```

      I8=4<9+K-D0      ; EQ 15:
      RS  W2      H1.      ;
      BZ  W1      X3+3    ;
      RS  W1      H2.      ;
      JL  W2      J3.      ;   F:= EQ 4;
      RS  W0      H3.      ;
      BZ  W3      X2+2    ;
      JL  W2      C4.      ;
      RS  W0      H4.      ;   T1:= VALUE(STORE[ENTRY + 2]);
      RL  W3      H2.      ;
      JL  W2      C4.      ;   T2:= VALUE(STORE[ENTRY + 3]);
      WS  W0      H4.      ;

```

```

WM. W0      H3.      ;
WD. W0      G11.     ;   E:= F x (T2 - T1) / 100;
JL.         (H1.)    ;   RETURN;

```

```

; EQ 16 , PARAM: C1, D
; E = AxC1xPOLY(D-1300,L6-35)x2^(-20)

```

```

I9=3<9+K-D0      ;
AL W1            0      ; EQ16:
AM              -1      ;   A:= TRUE; B:= FALSE; GO TO IN AD;

```

```

; EQ 10 , PARAM: C1, D, T
; F = AxC1xPOLY(D-1300,T-35)x2^(-20)

```

```

I7=4<9+K-D0      ; EQ 10:
AM              -2      ;   A:= B:= FALSE; GO TO INAD;

```

```

; EQ 17 , PARAM: C1, D, T
; F = POLY(D-1300, T-35)

```

```

I10 = 4<9+K-D0      ; EQ 17:
AL W0              1      ;
AD W1             -1      ;   A:= FALSE; B:= TRUE;
HS. W0            H8.     ;
RS. W1            H5.     ; IN AD:
RS. W2            H0.     ;
RS. W3            H3.     ;
BZ W0             X3      ;
SO W3             , H8: 0   ;   IF -, B THEN
JL. W2            C2.     ;   A:= AD(STORE[ENTRY]);
RS. W0            H4.     ;
AM.              (H3.)    ;
BZ W3             2      ;
JL. W2            C4.     ;   DC:= VALUE(STORE[ENTRY + 2] - 1300;
BS. W0            G15.    ;
RS. W0            H2.     ;
AM.              (H3.)    ;
BZ W3             3      ;
AM               (B29)    ;
RL W0             E2+10   ;   T:= L6;
SO. W3            (H5.)   ;   IF -, A THEN
JL. W2            C4.     ;   T:= VALUE(STORE[ENTRY + 3]);
BS. W0            G16.    ;
RS. W0            H1.     ;
JL. W3            C5.     ;   E:= POLY(DC, T-35);
BL. W2            H8.     ;
SN W2             0      ;   IF B THEN
JL.              (H0.)    ;   RETURN;
WM. W0            H4.     ;

```

H
W

```
AM.          (H3.) ;  
BZ  W1       1   ;  
WM  W0       2   ;  
AD  W0      -20  ;   E:= E x A x C1 x 2 ^ (-20);  
JL.          (H0.) ;   RETURN;
```

H G15: 1300 , G16: 35 ;

```

B      H4, J6      ;
W      : EQ 0, PARAM : C1
      : E = C1 x A

      I4=2<9+K-D0      ; EQ 0:
      AM      1      ;   ADD:= FALSE; GO TO MULT;

      : EQ 00 , PARAM: C1, C2
      : E = C1 x A + C2

```

```

      I0=3<9+K-D0      ; EQ 00:
      AL W0      -1      ;   ADD:= TRUE;
      HS. W0      H1.      ;
      RS. W2      H0.      ; MULT:
      BZ W0      X3      ;
      JL. W2      C2.      ;   A:= AD(STORE[ENTRY]);
      BZ W1      X2+1      ;
      WM W0      2      ;   K:= A x C1;
      AD W0      -10      ;   E:= K x 2 ^ (-10);
H      SO W2      , H1: 0      ;   IF ADD THEN
W      RA W0      X2+2      ;   E:= E + C2;
      JL.      (H0.)      ;   RETURN;

```

```

W      : CONSTANTS FOR POLYNOMIALS

```

```

J6:
J0:      -625 938      ; K0 FOR EQ 11
      95 690      ; K1
      715      ; K2
J1:      -68      ; K0 FOR EQ 12
      65 118      ; K1
      -8 102      ; K2
      4 378      ; K3
      - 812      ; K4
J2:      -33      ; K0 FOR EQ 13
      79 202      ; K1
      5 088      ; K2
      -5 594      ; K3
      1 876      ; K4
      - 194      ; K5
J3:      312      ; K0 FOR EQ 14
      213 330      ; K1
      -155 191      ; K2
      138 377      ; K3
      -69 245      ; K4
      17 322      ; K5
      -1 690      ; K6
J4:      0      ; K0 FOR EQ 6
      0      ; K1
      141 740      ; K2

```

-26 230 ; K3

J5:

; REDEFINE CONSTANT NAMES

J0=J0-J6, J1=J1-J6, J2=J2-J6, J3=J3-J6
 J4=J4-J6, J5=J5-J6

```

; EQ 6:
I11=1<9+K-D0 ; FIRST:= J4; LAST:= J5; GO TO A;
AM J4<6-J3<6+J5-J4 ;
; EQ 14:
I15=1<9+K-D0 ; FIRST:= J3; LAST:= J4; GO TO A;
AM J3<6-J2<6+J4-J3 ;
; EQ 13:
I14=1<9+K-D0 ; FIRST:= J2; LAST:= J3; GO TO A;
AM J2<6-J1<6+J3-J2 ;
; EQ 12:
I13=1<9+K-D0 ;
AL W0 J1<6+J2 ; FIRST:= J1; LAST:= J2;
LD W1 -6 ; A:
LS W1 -18 ;
HS. W0 H2. ;
RS. W2 H0. ;
RX W3 2 ;
BZ W0 X1 ;
JL. W2 C2. ; A:= AD(STORE[ENTRY]);
H3: AL W1 (0) ; CYCLE:
AL W0 0 ; E:= 0;
H4: AL W2 X2-2 ; FOR K:= LAST - 2 STEP -2 UNTIL FIRST DO
WM W0 2 ; E:= (E * A) SHIFT (-8) + CONST[K];
AD W0 -8 ;
WA. W0 X2+J6. ;
SE W2 , H2: 0 ;
JL. H4. ;
AS W0 -8 ; E:= E SHIFT (-8);
JL. (H0.) ; RETURN;

I12=1<9+K-D0 ; EQ 11:
RS. W2 H0. ;
RZ W0 X3 ;
JL. W2 C2. ; A:= AD(STORE[ENTRY]);
HS. W3 H2. ; FIRST:= J0;
AL W2 -23 ;
WM W2 0 ; K:= -23 * A;
WM. W0 G7. ;
WA. W0 G8. ;
AS W0 4 ; L:= (12 * A + 10000) * 2 ^ 4;
WM. W0 G9. ;
WA. W2 G10. ;

```

H
W

```
WD  W0      4      ;  A:= L x 1000 / (100000 + K);  
AL  W2      J1      ;  LAST:= J1;  
JL.      H3.      ;  GO TO CYCLE;
```

```
G7:      12      ;  CONSTANTS  
G8:      10 000   ;  
G9:      1 000    ;  
G10:     100 000  ;  
G11:     100      ;
```

```
H0:      0      ;  RETURN ADD. FOR EQ 00,0  
          ;  6, 11, 12, 13, 14
```

E

```
; THE FOLLOWING PROCEDURE INPUTS THE VALUE
; OF AN A/D TERMINAL, AND CONVERTS IT TO BINARY
; FORM. THE INPUT VALUE CONSIST OF A SIGN
; (1 STANDS FOR MINUS) AND THREE BCD DIGITS.
; THE CONVERSION IS OBTAINED BY SUCCESSIVE MULTIPLY-
; CATION WITH 10.
; THE INPUT IS REPEATED UP TO 5 TIMES. THE FIRST TIME
; TWO SUCCESSIVE INPUTS HAVE AN ABSOLUTE DIFFERENCE LESS THAN
; 5 THE LAST INPUT VALUE IS USED AS THE ACTUAL VALUE.
; BY THE ASSEMBLY CONDITION A56=I11 CAN BE CHOSEN ONLY TO
; INPUT THE VALUE ONCE.
; ERRORNEUS INPUT CAUSES AN ERROR MESSAGE AND PRODUCE THE
; VALUE 0. THIS AVOID BLOCKING THE SYSTEM.
; THE A/D TERMINALS CAN BE SIMULATED, IN
; THIS CASE THE VALUE OF TERMINAL CORRESPONDING
; TO SCAN NUMBER I IS STORED
; IN THE I-TH POSITION OF A TABLE IN SEGMENT 9.
```

```
; INTEGER PROCEDURE AD(TERMINAL);
; REG      TERMINAL/ AD
;          FREE / DESTROYED
;          RETURN/ SAVED W3
;          FREE / ZERO
; CALL:    JL. W2 C2.
; TESTOUTPUT: AD <RESULT>
```

```
C      A56<5, H8      ; IF NOT SIMULATED THEN
W C2:  RS. W2          H0. ; BEGIN
      RS. W3          H4. ;
C      A56<11         ; IF NOT ONCE THEN
W      RL. W2          G5. ; BEGIN PREVIOUS:=BIG;
      AL W3           5    ; I:=5;
H7:    AL W3          X3 -1 ; NO MATCH: I:=I-1;
      SH W3           0    ; IF I<1 THEN
      JL.            H5. ; GOTO MESSAGE;
E      ; END;
H      A50 , A31 ;
      H1. , A21+A6 ;
      A50 , A31 ; M:= AD INPUT(TERMINAL);
      H1. , A20+A6 ;
C      A56<11         ; IF NOT ONCE THEN
W      AL W1          X2  ; BEGIN
      WS. W1          H1. ; IF ABS(PREVIOUS - M) > 5
      RL. W2          H1. ; THEN BEGIN
      SH W1           5    ; PREVIOUS:=M; GOTO NO MATCH
      SH W1          -6    ; END;
      JL.            H7. ;
E      ; END;
W      RL. W1          H1. ;
      SZ. W1          (G5.) ; IF ERROR THEN
      JL.            H5. ; GOTO MESSAGE;
      RL. W2          G5. ;
      LD W2          -12 ;
      HS. W1          H3. ;
      AL W0           0    ; L:= 0;
H2:    WM. W0          G4. ; FOR K:= 15 STEP 4 UNTIL 23 DO
      AL W1           0    ; L:= 10 * L + BITS(K-4, K)
      LD W2           4    ; OF: (M);
      WA W0           2    ;
      SE. W2          (G5.) ;
```

W
C
H
E

AC W0 (0) ; L:= -L; AD:= L;
[A58>18~1] - 1 ; IF TESTOUTPUT THEN
A51 , <AD> ; TESTOUT(<AD>, L);

H6: RL. W2 H4. ; END:


```

      JL.          (H0.) ; RETURN;
H5:   AL  W1       A6   ; MESSAGE:
      JL  W3       (B21) ; ERRORMESSAGE(AD, TERMINAL);
      LD  W0       -48   ; AD:= 0;
      JL.          H6.   ; GO TO END;
H0:   ;           0     ; WORKING LOCATIONS
H1:   ;           0     ; FOR STATUS
H4:   ;           0     ; SAVE W3
E     ;           ; END
C     ;           ; ELSE
W C2: RS. W0       H4.   ; BEGIN
      RS. W3       H1.   ;
      RS. W2       H0.   ;
      AL  W3       0     ; LAST:= 0;
      HS. W3       H2.   ; I:= 0;
      ;           ; NEXT:
H3:   JL. W2       C3.   ; LAST:= NEXT ENTRY(LAST);
      SH  W0       -1    ; IF GROUP ~ END ~ CR THEN
      JL.          H3.   ; GO TO NEXT;
H     AL  W1       , H2: 0 ;
W     AL  W1       X1+1   ; I:= I + 1;
      HS. W1       H2.   ;
      AL  W2       X3     ; J:= LAST;
      SZ  W0 000100 000000 ; IF BIT L THEN
      AL  W2       X2+1   ; J:= J + 1;
      SZ  W0 001000 000000 ; IF BIT H THEN
      AL  W2       X2+1   ; J:= J + 1;
      AM          (B30)   ;
      BL  W2       X2+1   ; IF AD TABLE[J+1]=TERMINAL
      SE  W0       16     ; ~ EQPART=DUMMY THEN
      SE. W2       (H4.)  ;
      JL.          H3.   ; GO TO NEXT;
      AM          (B18)   ;
      BL  W0       X1-1   ; AD:= SIMULATION TABLE[I-1];
      AL  W3       0     ;
      RL. W2       H1.   ;
      A50 , A37 ; GIVE A TURN;
      [A58>18~1] - 1 ; IF TESTOUTPUT THEN
      A51 , <AS> ; TESTOUT(<AS>, AD);
H     JL.          (H0.) ; END;
C     ;           ;
W     H0:         0     ;
      H1:         0     ;
      H4:         0     ;
E     ;           ;
W G4:         10     ;
      G5:         7<20 ;

```

```

; INTEGER PROCEDURE SQRT(X);
; REG   W0   X   / IF X < 2 THEN X ELSE SQRT(X)
;       W1   FREE / DESTROYED
;       W2   SAVED
;       W3   LINK
; CALL   JL. W3 C1.

```

```

R H1, J4
W C1: SH W0      1      ; IF X < 2 THEN
      JL      X3      ;   SQRT:= X ELSE
      RS. W3      J2.   ;   BEGIN
      NS. W0      3      ;   C:= 0; X:= X + 2 ^ (-23);
      AL W1      0      ;   FOR C:= C, C-1 WHILE X < 1 DO X:= 2 * X;
      SO W1      1      ;   IF -, ODD(C) THEN
      LS W0      -1     ;       X:= X / 2;
      LD W1      -1     ;   X:= X / 2 ; C:= C : 2;
      RS. W0      J3.   ;
      WA. W0      G17.  ;   Y[C]:= X + 0.5;
      HS. W1      J1.   ;
      AL W1      10000  ;

H1: RS. W0      J4.   ; FOR I:= 1 STEP 1 UNTIL 4 DO
      AL W0      0      ;
      RL. W3      J3.   ;
      WD. W0      J4.   ;   Y[I]:= (Y[I-1] + 2 * X / Y[I-1]) / 2;
      WA. W0      J4.   ;
      LD W1      -1     ;
      SO W1      00001  ;
      JL.      H1.   ;
      AM      -10     ;
H   LS W0      , J1: 0   ; INTEGER C;
W   BS. W0      G13.  ; SQRT:= (Y[4] * 2 ^ (C-10) * 2 ^ 24
H   LS W0      , G13:-1 ;   + 1)/2;
W   JL.      (J2.)  ; END;

J2:      0      ;
J3:      0      ; X
J4:      0      ; Y[I]
G17:     1<22  ; 0.5 * 2 ^ 23
E

```

```

; INTEGER PROCEDURE POLY(TC, DC);
; REG   W0   - / POLY
;       W1   DESTROYED
;       W2   DESTROYED
;       W3   LINK
; CALL   JL. W3 C5. , TC IN H1 , DC IN H2
B      I5, J2      ; BEGIN
W C5:  RS. W3      I0. ;
      AL  W1      0   ; VALUE:=0;
      AL. W2      J1. ;
      I2: WM. W1   H1. ;
      AD  W1      -6  ;
      RS. W1      I1. ;
      AL  W3      X2+8 ; FOR I:= 1 STEP 1 UNTIL 3 DO
      AL  W1      0   ; BEGIN W1:= 0;
      I3: WM. W1   H2. ; FOR J:= 1 STEP 1 UNTIL 4 DO
      AD  W1      -6  ; W1:= W1 x DC SHIFT (-6)
      WA  W1      X2   ; + K [I, J];
      AL  W2      X2+2 ;
      SE  W2      X3   ;
      JL.        I3. ;
      WA. W1      I1. ; VALUE:=VALUE*TC SHIFT (-6)+W1;
      SE. W2      J2. ;
      JL.        I2. ; END;
      AL  W0      X1   ;
      AS  W0      -6   ; POLY:= VALUE SHIFT (-6)
      JL.        (I0.) ; END;

```

```

; CONSTANTS FOR POLYNOMIAL:

```

```

J1:      -2 029      ; INTEGER ARRAY K[1:3, 1:4];
        -4 775      ;
        -1 568      ;
         3 423      ;
        -1 122      ;
        -1 474      ;
         2 576      ;
         9 089      ;
         136        ;
         473        ;
         7 143      ;
        32 825      ;

J2:

I0:      0          ;
I1:      0          ; INTEGER VALUE;

```

```

E
E

```

```

B      E0.      ; TAIL SEGMENT 5:
W E0:  AL. W0    C0.      ; INITIALIZE(
        RS W0    R20      ; VALUE,
        AL. W0    C3.      ;
        RS W0    R23      ; NEXT ENTRY,
        AL. W0    E0.      ; FIRST FREE);
        AL W1     0        ;
        AL W2     A54>18^1 ;
        JL W3     (B39)    ; CALL LOADER;

        JL.      E0.      ;

```

```

E
L
; IDLIST S5
N
T
E
E
L
; END S5
N
T

```


C [A55>17^1]-1
S E1 , K=0

; SEGMENT 6:

; THE SEGMENT CONTAINS THE MACROPROGRAM TO BE
; EXECUTED BY SEGMENT 7 DURING BALANCE CALCULATION, THE
; MACROINSTRUCTIONS ARE GIVEN BOTH IN SYMBOLIC AND
; ASSEMBLED FORM. THE TRANSLATION WAS MADE BY AN OFF-LINE
; PROGRAM. A MACROINSTRUCTION BYTE IS INTERPRETED IN THE
; FOLLOWING WAY:

; BITS IN A BYTE: 1 1 1 1 1 1 1 1 1 1 1
; [ADDRESS][INST]

; THE POSSIBLE MACROS AND THEIR VALUES:

; TR, AR, TS, ST, SW, MU, DI, TF, AF, TC, EX, TM, AM
; 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

; THE LETTERS L, C, W AND E CAUSE THE ADDITION OF A CONSTANT
; TO THE ADDRESS PART:

; L = E2:2 , C = L + E3:2 , W = 125 , E = 109

; THE FINAL ADDRESS FOR THE MACROS TS, TF, AF, TC, TM
; AND AM WAS OBTAINED DURING ASSEMBLY BY SEARCHING VIA CROSS-
; REFERENCE TABLES. INFORMATION FOR MAKING THESE TABLES IS
; SUPPLIED BY SEGMENT 11.

H E0:

; 1.	TF	1	, AF	2	, AF	3	, ST	0
		7	, 24	, 40	, 3			
; 2.	TR	1	, MUL	1	, DIC	1	, ST	0
		16	, 2181	, 2278	, 3			
; 3.	TS	9						
		130						
; 4.	TS	10						
		210						
; 5.	TS	11						
		290						
; 6.	TR	3	, AR	4	, AR	5	, SWW	1
		48	, 65	, 81	, 2020			
; 7.	TR	3	, MU	1	, DIW	1	, ST	0
		48	, 21	, 2022	, 3			
; 8.	TR	4	, MU	1	, DIW	1	, ST	0
		64	, 21	, 2022	, 3			
; 9.	TR	5	, MU	1	, DIW	1	, ST	0
		80	, 21	, 2022	, 3			
; 10.	TC	30	, MUL	2	, DIC	3	, SWW	1
		489	, 2197	, 2310	, 2020			
; 11.	TC	33	, MUL	3	, DIC	2	, ARW	1
		505	, 2213	, 2294	, 2017			
; 12.	DIC	4	, AF	36	, ST	0		
		2326	, 520	, 3				
; 13.	TC	31	, MUL	2	, DIC	3	, SWW	1
		553	, 2197	, 2310	, 2020			
; 14.	TC	34	, MUL	3	, DIC	2	, ARW	1
		569	, 2213	, 2294	, 2017			
; 15.	DIC	4	, AF	37	, ST	0		
		2326	, 584	, 3				
; 16.	TC	32	, MUL	2	, DIC	3	, SWW	1
		617	, 2197	, 2310	, 2020			
; 17.	TC	35	, MUL	3	, DIC	2	, ARW	1
		633	, 2213	, 2294	, 2017			
; 18.	DIC	4	, AF	38	, ST	0		
		2326	, 648	, 3				
; 19.	TR	9	, AR	10	, AR	11	, ST	0
		144	, 161	, 177	, 3			
; 20.	TR	2	, MUC	4	, DI	12	, ST	0
		32	, 2325	, 198	, 3			
; 21.	TF	42	, MUL	4	, DIC	4	, AF	41
		711	, 2229	, 2326	, 696			
; 22.	ST	0						
		3						
; 23.	TR	20	, AR	21	, AR	22	, MUL	5
		320	, 337	, 353	, 2245			
; 24.	DIC	4	, ST	0				
		2326	, 3					
; 25.	TS	18						
		178						
; 26.	TS	19						
		258						
; 27.	TS	20						
		338						
; 28.	TF	15	, DIC	3	, ST	0		
		167	, 2310	, 3				
; 29.	TF	16	, DIC	3	, ST	0		
		247	, 2310	, 3				
; 30.	TF	17	, DIC	3	, ST	0		
		327	, 2310	, 3				
; 31.	TR	16	, AR	19	, ST	0		
		256	, 305	, 3				
; 32.	TR	17	, AR	20	, ST	0		
		272	, 321	, 3				

; 25.	TR	22 , AR 23 , AR 24 , ST 0	352 , 369 , 385 , 3
; 26.	TS	12	146
; 27.	TS	13	226
; 28.	TS	14	306
; 29.	TR	19 , AR 20 , AR 21 , AR 26	304 , 321 , 337 , 417
; 30.	TS	27 , AR 28 , ST 0	433 , 449 , 3
; 31.	TS	27	466
; 32.	TS	28	530
; 33.	TS	29	594
; 34.	TR	30 , MUC 4 , DI 9 , ST 0	480 , 2325 , 150 , 3
; 35.	TR	31 , MUC 4 , DI 10 , ST 0	496 , 2325 , 166 , 3
; 36.	TR	32 , MUC 4 , DI 11 , ST 0	512 , 2325 , 182 , 3
; 37.	TS	39	658
; 38.	TS	40	674
; 39.	TS	21	370
; 40.	TS	22	402
; 41.	TS	23	434
; 42.	TS	24	386
; 43.	TS	25	418
; 44.	TS	26	450
; 45.	TR	41 , AR 42 , AR 43 , ST 0	656 , 673 , 689 , 3
; 46.	TS	49	818
; 47.	TS	71	930
; 48.	TS	70	914
; 49.	TS	47	786
; 50.	TS	48	802
; 51.	TS	43	722
; 52.	TS	44	738
; 53.	TS	45	754
; 54.	TS	46	770
; 55.	TC	50 , ST 0	841 , 3
; 56.	TC	51 , ST 0	953 , 3
; 57.	TC	52 , ST 0	1033 , 3
; 58.	TC	53 , ST 0	1113 , 3
; 59.	TR	54 , AR 55 , AR 56 , AR 57	864 , 881 , 897 , 913

; 59.	TS	66					
		898					
; 60.	TS	67					
		1010					
; 61.	TS	68					
		1090					
; 62.	TS	69					
		1170					
; 63.	TR	59	, AR	60	, AR	61	, AR 62
		944	, 961	, 977	, 993		
; 64.	TR	54	, MUC	2	, DI	59	, ST 0
		864	, 2293	, 950	, 3		
; 65.	TR	55	, MUC	2	, DI	60	, ST 0
		880	, 2293	, 966	, 3		
; 66.	TR	56	, MUC	2	, DI	61	, ST 0
		896	, 2293	, 982	, 3		
; 67.	TR	57	, MUC	2	, DI	62	, ST 0
		912	, 2293	, 998	, 3		
; 68.	TS	54					
		850					
; 69.	TS	55					
		962					
; 70.	TS	56					
		1042					
; 71.	TS	57					
		1122					
; 72.	TR	68	, MUC	4	, DI	59	, ST 0
		1088	, 2325	, 950	, 3		
; 73.	TR	69	, MUC	4	, DI	60	, ST 0
		1104	, 2325	, 966	, 3		
; 74.	TR	70	, MUC	4	, DI	61	, ST 0
		1120	, 2325	, 982	, 3		
; 75.	TR	71	, MUC	4	, DI	62	, ST 0
		1136	, 2325	, 998	, 3		
; 76.	TS	62					
		882					
; 77.	TS	63					
		994					
; 78.	TS	64					
		1074					
; 79.	TS	65					
		1154					
; 80.	TR	76	, AR	77	, AR	78	, AR 79
		1216	, 1233	, 1249	, 1265		
; 81.	TS	58					
		866					
; 82.	TS	59					
		978					
; 83.	TS	60					
		1058					
; 84.	TS	61					
		1138					
; 85.	TR	81	, AR	82	, AR	83	, AR 84
		1296	, 1313	, 1329	, 1345		
; 86.	TF	72	, AF	75	, ST	0	
		1191	, 1208	, 3			
; 87.	TF	73	, AF	76	, ST	0	
		1271	, 1288	, 3			
; 88.	TF	74	, AF	77	, ST	0	
		1351	, 1368	, 3			
; 89.	TR	86	, AR	87	, AR	88	, ST 0
		1376	, 1393	, 1409	, 3		
; 90.	TF	78	, AF	81	, ST	0	
		1223	, 1240	, 3			

; 92.	TF	80	, AF	83	, ST	0	
		1383	, AR	1400	, ST	3	
; 93.	TR	90	, AR	91	, AR	92	0
		1440	, AR	1457	, AR	1473	3
; 94.	TM	60	, AM	61	, AM	62	63
		939	, AM	956	, AM	972	988
; 95.	AM	64	, AM	65	, AM	66	67
		1004	, AM	1020	, AM	1036	1052
; 96.	ST	0					
		3					
; 97.	TR	2	, MUC	4	, DI	94	0
		32	, MUC	2325	, DI	1510	3
; 98.	TR	89	, MUC	2	, DI	94	0
		1424	, MUC	2293	, DI	1510	3
; 99.	TR	93	, MUC	4	, DI	94	0
		1488	, MUC	2325	, DI	1510	3
; 100.	TS	84					
		1250					
; 101.	TS	85					
		1330					
; 102.	TS	86					
		1410					
; 103.	TS	5					
		66					
; 104.	TS	8					
		114					
; 105.	TR	101	, MUC	4	, SWW	1	102
		1616	, MUC	2325	, SWW	2020	1632
; 106.	MUC	2	, ARW	1	, DI	94	0
		2293	, ARW	2017	, DI	1510	3
; 107.	TS	6					
		82					
; 108.	TS	4					
		50					
; 109.	TS	7					
		98					
; 110.	TS	87					
		194					
; 111.	TS	88					
		274					
; 112.	TS	89					
		354					
; 113.	TR	3	, AR	4	, AR	5	SWW 1
		48	, AR	65	, AR	81	2020
; 114.	TM	2	, AM	3	, AM	4	SWW 2
		11	, AM	28	, AM	44	2036
; 115.	MU	3	, DIW	1	, AM	5	AM 8
		53	, DIW	2022	, AM	60	252
; 116.	AM	9	, AM	14	, ST	0	
		300	, AM	348	, ST	3	
; 117.	TR	4	, MUW	2	, DIW	1	AM 6
		64	, MUW	2037	, DIW	2022	76
; 118.	AM	10	, AM	11	, AM	15	ST 0
		268	, AM	316	, AM	364	3
; 119.	TR	5	, MUW	2	, DIW	1	AM 7
		80	, MUW	2037	, DIW	2022	92
; 120.	AM	12	, AM	13	, AM	16	ST 0
		284	, AM	332	, AM	380	3
; 121.	TRE	1	, ARE	2	, ARE	3	ST 0
		1760	, ARE	1777	, ARE	1793	3
; 122.	TM	17	, AM	20	, AM	21	ARE 1
		107	, AM	156	, AM	204	1761
; 123.	ST	0					
		3					
; 124.	TM	18	, AM	22	, AM	23	ARE 2
		123	, AM	172	, AM	220	1777
; 125.	ST	0					
		3					
; 126.	TM	19	, AM	24	, AM	25	ARE 3
		139	, AM	188	, AM	236	1793

; 8.	TRE	5	, ARE	6	, ARE	7	, ST	0
		1824	, 1841	, 1857	, 1857	, 1857	, 1857	3
; 9.	TRE	4	, MUC	4	, DI	12	, ST	0
		1808	, 2325	, 198	, 198	, 198	, 198	3
; 10.	TM	33	, MU	59	, DI	63	, AM	29
		811	, 949	, 1014	, 1014	, 1014	, 1014	748
	AM	34	, ST	0				
		652	, 3					
; 11.	TM	33	, MU	60	, DI	63	, AM	30
		811	, 965	, 1014	, 1014	, 1014	, 1014	764
	AM	35	, ST	0				
		668	, 3					
; 12.	TM	33	, MU	61	, DI	63	, AM	31
		811	, 981	, 1014	, 1014	, 1014	, 1014	780
	AM	36	, ST	0				
		684	, 3					
; 13.	TM	33	, MU	62	, DI	63	, AM	32
		811	, 997	, 1014	, 1014	, 1014	, 1014	796
	AM	37	, ST	0				
		700	, 3					
; 14.	TRE	10	, ARE	11	, ARE	12	, ARE	13
		1904	, 1921	, 1937	, 1937	, 1937	, 1937	1953
	ST	0						
		3						
; 15.	TM	39	, AM	40	, ARE	14	, ST	0
		715	, 732	, 1969	, 1969	, 1969	, 1969	3
; 16.	TRE	14	, MUC	4	, DI	63	, ST	0
		1968	, 2325	, 1014	, 1014	, 1014	, 1014	3
; 17.	TM	41	, ST	0				
		843	, 3					
; 18.	TM	42	, ST	0				
		859	, 3					
; 19.	TM	43	, ST	0				
		875	, 3					
; 20.	TRE	17	, ARE	18	, ARE	19	, ST	0
		2016	, 2033	, 2049	, 2049	, 2049	, 2049	3
; 21.	TM	44	, AM	45	, ARE	20	, ST	0
		891	, 908	, 2065	, 2065	, 2065	, 2065	3
; 22.	TRE	20	, MUC	4	, DI	94	, ST	0
		2064	, 2325	, 1510	, 1510	, 1510	, 1510	3
; 23.	TRE	4	, ARE	14	, ARE	20	, MUC	4
		1808	, 1969	, 2065	, 2065	, 2065	, 2065	2325
	DI	94	, ST	0				
		1510	, 3					
; 24.	TM	47	, AM	48	, AM	49	, AM	50
		443	, 460	, 476	, 476	, 476	, 476	492
	AM	51	, ST	0				
		508	, 3					
; 25.	TM	52	, AM	53	, AM	54	, AM	55
		523	, 540	, 556	, 556	, 556	, 556	572
	AM	56	, AM	57	, AM	58	, ST	0
		588	, 604	, 620	, 620	, 620	, 620	3
; 26.	TM	26	, AM	27	, AM	28	, AM	38
		395	, 412	, 428	, 428	, 428	, 428	828
	AM	46	, AM	59	, ST	0		
		924	, 636	, 3				
	EX	0						
		10						

```

; TAIL SEGMENT 6:
W E1:  AL. W0      E0,-1 ;  INITIALIZE(
        RS W0      B22   ;      B22));
        AL. W0      E1.   ;
        AL W1       0     ;
        AL W2      A54>17^1 ;
        JL W3      (B39)  ;  CALL LOADER;

        JL.         E1.   ;

```

```

E
E
L
; END S6
N
T

```


C [A55>16^1] - 1
S D5, G5, H10, I7, J15
W
K=0

```
; SEGMENT 7:
; THE FOLLOWING PROGRAM PERFORMS THE CALCULATION
; FOR THE 8 HOURLY BALANCE REPORT.
; THE CALCULATION IS DONE BY SIMULATING A
; SUITABLE PROCESSOR. THIS NON-EXISTENT COMPUTER
; HAS 8 BITS ADDRESS AND 4 BITS INSTRUCTION PART.
; THE DECODED INSTRUCTIONS AND THEIR MEANING IS
; GIVEN BELOW. THE MACROINSTRUCTIONS DESCRIBING
; THE CALCULATION FORM SEGMENT 6.
; THE PROGRAM BEGINS WITH INPUTING SOME
; A/D VALUES, WHICH ARE USED IN FORMULAE 15. (SEE SEGMENT 6)
; OVERFLOW IS CHECKED AT DIVISION, THUS IF OVERFLOW
; OCCURS, THE RESULT IS SET EQUAL TO A SPECIAL NONSENSE
; VALUE. IF SUCH A VALUE HAPPENS TO BE AN OPERAND OF A
; MACORINSTRUCTION, THE RESULT WILL BE THE SAME. (THE NONSENSE)
; ON THE LOG SHEET NONSENSE WILL APPEAR AS 0000-
; TEST OUTPUT:
;           AC    <VALUE OF THE ACCUMULATOR>
;           IN    <INSTRUCTION CODE>
;           AD    <ADDRESS>
; MACROINSTRUCTION MNEMONICS AND THEIR MEANING:
;
; TR      TAKE R
; AR      ADD  R
; TS      TAKE F AND STORE
; ST      STORE
; SW      STORE IN WORKING LOCATION
; MU      MULTIPLY R
; DI      DIVIDE R
; TF      TAKE F
; AF      ADD F
; TC      TAKE F AND CORRECT BY 1703/2240
; EX      EXIT MACROPROGRAM
; TM      TAKE M
; AM      ADD  M
;
; CONSTANTS, PLANT DATA AND WORKING LOCATIONS ARE
; TREATED AS R VALUES (EXCEPT STORE)
```



```

      JL  W2      (B20) ; R22:= VALUE(H3);
      RS. W0      D2,+42 ;
      RL  W1      B22  ; CC:= FIRST MACROINSTRUCTION - 1
H0:    ; FETCH:
      AL  W1      X1+1  ; CC:= CC + 1;
      BZ  W2      X1    ;
      LD  W3      -4    ; AD:= ADDRESS PART (STORE[CC]) * 2;
      LS  W3      -20   ; IN:= INSTRPART(STORE[CC]);
      WA  W2      4     ;
C      [A58>16^1] - 1 ; IF TESTOUTPUT THEN
      ; BEGIN
H      A51      ,    <AC> ; TESTOUT(<AC>, ACC);
W      RX  W0      6     ;
H      A51      ,    <IN> ; TESTOUT(<IN>, IN);
W      RX  W0      4     ;
H      A51      ,    <AD> ; TESTOUT(<AD>, AD)
W      RX  W0      6     ;
E      RX  W2      6     ; END;

      BL. W3      X3+H1. ; GO TO INSTRUCTION[IN];
H2:    JL. W3      X3+H2. ;
      WA  W0      4     ;
      JL.          H0.  ;

J0:    AL  W0      0     ; TR:
      RS. W0      10.   ; LONGACC:= 0;
J1:    ; AR:
      RL. W2      X2+D2.-2 ; K:= READJ;
      SN. W2      (I1.) ; IF K = NONSENSE THEN
      RS. W2      I3.   ; OVERFLOW:= K;
      JL          X3    ; ACC:= ACC + K
      ; GO TO FETCH;
J2:    ; TS:
      JL. W3      H3.   ; ACC:= FVALUE;
J3:    RL. W3      I3.   ; ST:
      SN. W3      (I1.) ; IF OVERFLOW = NONSENSE THEN
      AL  W0      X3    ; ACC:= NONSENSE;
      RS. W1      I3.   ; OVERFLOW:= CC;
H      AL  W2      , I6: 0 ;
W      AL  W2      X2+2  ; INDEX:= AD:= INDEX + 2;
      HS. W2      I6.   ;
J4:    ; SW:
      RS. W0      X2+D2.-2 ; READD:= ACC;
      JL.          H0.   ; GO TO FETCH;

J5:    ; MU:
      JL. W3      J1.   ; K:= READJ; IF K = NONSENSE THEN
      ; OVERFLOW := K;
      WM  W0      4     ;

```



```

RS. W3      IO.      ; LONGACC:= ACC * K;
JL.         HO.      ; GO TO FETCH;

J6:         ; DI:
JL. W3      J1.      ; K:= R[AD]; IF K = NONSENSE THEN
                ; OVERFLOW := K;
RL. W3      IO.      ;
H H5: AD W0   , G2: 1  ; DIVIDE LONG:
W WD W0      4        ; K:= LONGACC * 2 : K;
AL. W2      (I1.)    ;
SX          10       ; IF ABS(K) > 2 ^ 24 - 1 THEN
RS. W2      I3.      ; OVERFLOW:= NONSENSE;
BA. W0      G2.      ;
AS W0      -1        ; ACC:= (K+1) : 2;
JL.         HO.      ; GO TO FETCH;
J7: AL W0      0      ; TF:
RS. W0      IO.      ; LONGACC:= 0;
J8: AL W3      HO.      ; AF:
                ; ACC:= ACC + FVALUE;
JL.         H4.      ; GO TO FETCH;

J9: JL W3      H3.      ; TC:
WM. W0      G1.      ; LONGACC:= FVALUE * 1703;
RL. W2      G0.      ; K:= 2240;
JL.         H5.      ; GO TO DIVIDE LONG;

H3: AL W0      0      ; INTEGER PROCEDURE F VALUE;
H4: RS. W0      I2.      ;
RL. W0      X2+D0.    ; FVALUE:= ENTIER(F[AD] *
AL W2      X3        ; PERIOD /
WM W0      B46       ;
WD. W0      I7.      ;
AL W3      0         ; 3600/NO OF STEPS);
WD W0      B40       ;
WA. W0      I2.      ;
JL         X2        ;

J10:         ; EX:
JL         (B47)    ; GO TO RESET PHASE;
J11:         ; TM:
AL W0      0        ; ACC:=0;
J12:         ; AM:
RL. W2      X2+D1.    ; ACC:=ACC+M[AD]:2;
AS W2      -1        ;
JL.         H2.+2    ; GO TO FETCH;

                ; SWITCH INSTRUCTION:=

```

```
H H1:      J0-H2      ;      TR,
            J1-H2      ;      AR,
            J2-H2      ;      TS,
            J3-H2      ;      ST,
            J4-H2      ;      SW,
            J5-H2      ;      MU,
            J6-H2      ;      DI,
            J7-H2      ;      TF,
            J8-H2      ;      AF,
            J9-H2      ;      TC,
            J10-H2     ;      EX,
            J11-H2     ;      TM,
            J12-H2     ;      AM;

W G0:      2240      ;
G1:      1703      ;
I1:      -1 000 000 ;      NONSENSE

I0:      0          ;      INTEGER LONG ACC,
I2:      0          ;
I3:      0          ;      OVERFLOW;
I7:      3600      ;
```

```
B E0
W E0:  AL, W0      D3, ; TAIL SEGMENT 7:
        RS W0      B44 ; INITIALIZE(
        AL, W0      D2, ;
        RS W0      B29 ; R TABLE
        AL, W0      D0, ;
        RS W0      B34 ; F TABLE.
        AL, W0      D1, ;
        RS W0      B35 ; M TABLE,

        AL, W0      E0, ; FIRST FREE);
        AL W1        0 ;
        AL W2      A54>16^1 ;
        JL W3      (B39) ; CALL LOADER;
        JL.         E0, ;
```

```
E
L
; IDLIST S7
N
T E
E
L
; END S7
N
T
```


c [a55>15^1]-1
s k=0, h20,i20

```

w h0:  r1  w3      b18  ; STEP:
        wa. w3      i0.  ;   begin
        bl  w1      x3-1 ;   simulation table[scan no]:=
h       al  w1      x1, i1: 0 ;   simulation table[scan no]+step;
w       hs  w1      x3-1 ;   exit task
h       a50      a32  ;   end;

w h2:  sl  w1      0      ; procedure test term;
        jl      x3      ;   if term<0 then
        sn. w1      (i3.) ;   go_to if term=<i> then INCREASE
        jl.       h5.    ;
        sn. w1      (i4.) ;   else if term=<d> then DIGITAL DIV
        jl.       h3.    ;
        sn. w1      (i5.) ;   else if term=<e> then EXIT
        jl.       h4.    ;   else SET;

h1:    al. w0      i2.    ; SET: writetext(<<scan no>);
        jl  w3      (b26) ;   i:=typein;
        jl  w3      (b27) ;
        jl. w3      h2.    ;   test term;
        rs. w0      i0.    ;
        al. w0      i7.    ;
        jl  w3      (b26) ;   writetext(<<value>);
        jl  w3      (b27) ;   j:=typein;
        jl. w3      h2.    ;   test term;
        rl  w3      b18    ;
        am.       (i0.)    ;
        hs  w0      x3-1   ;   simulation table[i]:=j;
        jl.       h1.    ;   go_to SET;

h4:    al. w0      i6.    ; EXIT:
        jl  w3      (b26) ;   writetext(<<exit>);
h       a50      a32    ;   exit task;

w h5:  al. w0      i8.    ; INCREASE:
        jl  w3      (b26) ;   writetext(<<increase by>);
        jl  w3      (b27) ;   step:=typein;
        jl. w3      h2.    ;   test term;
        hs. w0      i1.    ;
        jl.       h1.    ;   go_to SET;

h3:    al. w0      i9.    ; DIGITAL DEV:
        jl  w3      (b26) ;   writetext(<<digital device>);
        jl  w3      (b27) ;   i:=typein;
        jl. w3      h2.    ;   test term;
        ls  w0      1      ;
        rl  w2      0      ;   i:=base for digital devices+2*i;
        wa  w2      b18    ;
        al. w0      i11.   ;
        jl  w3      (b26) ;   writetext(<<bit>);
h6:    jl  w3      (b27) ;   NEXT BIT: j:=typein;
        jl. w3      h2.    ;   test term;
        hs. w1      h7.    ;
        rl  w3      0      ;
        sl  w0      0      ;   if j<0 then
        ac  w3      x3      ;   clear bit(-j)of:(store[i]) else
        al  w1      1      ;   set bit(j)of:(store[i]);
        ls  w1      x3+23  ;
        rl  w3      x2-32  ;

```

```

lo w3      2      ;
ac w1      x1+1   ;
sh w0      -1     ;
la w3      2      ;
rs w3      x2-32  ;
h w al w1   , h7: 0 ;
se w1      300    ;
jl.        h6.    ;
jl.        h1.    ;

```

go to if term=<> then
SET else NEXT BIT;

```

w i5: -3074
i4: -3016
i3: -3306
i0: 0 ; scan no
h i11: {bit <10>}
i2: {<63>scan no <10>}
i6: {<63>exit<10>}
i7: {value <10>}
i8: {<63>increase by <10>}
i9: {<63>digital device <10>}

```

```

w h10:
i12: {step}
i13: {set}
i14: 1<22
h11: rl. w0          i12. ; tail segment 8:
      rs w0          a46+6 ; initialise(
      rl. w0          i13. ; task 3,
      rs w0          a46+16; task 8);
      al. w0          h0. ;
      rs w0          a47+6 ;
      al. w0          h1. ;
      rs w0          a47+16;
      al w0          0 ;
      rs w0          a48+6 ;
      rl. w0          i14. ;
      rs w0          a48+16;
      al w0          10 ;
      rs w0          a49+16;

      al. w0          h10. ; call loader;
      al w1          0 ;
      al w2          a54>15^1
      jl w3          (b39) ;

      jl.            h11.

```

```
l; idlist S8  
n  
t  
e  
e  
e  
e  
l; end S8  
n  
t
```


C [A55 > 14 ^ 1] = 1
S H1
W K=0

; SEGMENT 9:

; THE SEGMENT CONTAINS A TABLE WITH SIMULATED

; AD-VALUES ORDERED AFTER INCREASING SCAN NUMBERS:

W 0,0,0,0,0,0,0 ; DIGITAL INPUT
H 0 ; LOAD AUX.

H0:

167	:	1	,	PR	4812 A	,	10 KP/CM2G
429	:	2	,	TR	4811	,	22 C
640	:	3	,	FR	4811	,	32000 NM3/H
640	:	4	,	FR	4812	,	16000 NM3/H
640	:	5	,	FR	4813	,	8000 NM3/H
850	:	6	,	PI	4817	,	34 KP/CM2G
41	:	7	,	TI	4801-6	,	40 C
650	:	8	,	PRC	4816	,	6.5 KP/CM2G
20	:	9	,	TI	4801-5	,	20 C
775	:	10	,	PR	1107	,	31 KP/CM2G
575	:	11	,	TR	1101	,	165 C
700	:	12	,	PRC	4804	,	14 KP/CM2G
213	:	13	,	TI	4801-2	,	198 C
617	:	14	,	FI	4801	,	16500 KG/H
640	:	15	,	FI	4804	,	32000 KG/H
775	:	16	,	PRC	4811	,	31 KP/CM2G
293	:	17	,	TI	4801-1	,	270 C
640	:	18	,	FI	4802	,	16800 KG/H
640	:	19	,	FI	4803	,	32000 KG/H
718	:	20	,	FI	4805	,	17500 KG/H
750	:	21	,	PR	4808	,	30 KG/CM2G
625	:	22	,	PRC	4801	,	2.5 KG/CM2G
625	:	23	,	FRC	1301	,	13440 NM3/H
640	:	24	,	FRC	1302	,	18400 NM3/H
660	:	25	,	FR	1304	,	18680 KG/H
700	:	26	,	PRC	1301	,	35 KG/CM2G
567	:	27	,	FR	1305	,	11290 KG/H
500	:	28	,	TRC			
603	:	29	,	FRC	1306	,	43300 KG/H
659	:	30	,	FRC	1308	,	8930 NM/H
250	:	31	,	TR	1302-2	,	550 C
640	:	32	,	PR	1306	,	32 KP/CM2G
455	:	33	,	TI	1301-8	,	550 C
498	:	34	,	TI	1301-9	,	600 C
583	:	35	,	TI	1301-10	,	700 C
654	:	36	,	TI	1301-11	,	785 C
654	:	37	,	TI	1301-19	,	785 C
455	:	38	,	TI	1301-38	,	550 C
498	:	39	,	TI	1301-39	,	600 C
583	:	40	,	TI	1301-40	,	700 C
654	:	41	,	TI	1301-41	,	785 C
654	:	42	,	TI	1301-49	,	785 C
524	:	43	,	TR	1302-3	,	1000 C
163	:	44	,	TI	1301-6	,	200 C
200	:	45	,	AR	1301	,	2.0 VOL PCT
250	:	46	,	TR	1302-1	,	550 C

387 ; 47 , TR 1302-4 , 785 C
304 ; 48 , AR 1302 , 9.13 VOL PCT
625 ; 49 , TR 1302-6 , 1150 C
498 ; 50 , TR 1302-8 , 964 C
300 ; 51 , AR 1303 , 0.30 VOL PCT
550 ; 52 , TRC 1306 , 360 C

625 ; 53 , FRC 1301 , 13440 NM3/H
640 ; 54 , FRC 1302 , 18400 NM3/H
660 ; 55 , FR 1304 , 18680 KG/H
700 ; 56 , PRC 1301 , 35 KG/CM2G
567 ; 57 , FR 1305 , 11290 KG/H

500 ; 58 , TRC
603 ; 59 , FRC 1306 , 43300 KG/H
659 ; 60 , FRC 1308 , 8930 NM/H

250 ; 61 , TR 1302-2 , 550 C
640 ; 62 , PR 1306 , 32 KP/CM2G

455 ; 63 , TI 1301-8 , 550 C

498 ; 64 , TI 1301-9 , 600 C

583 ; 65 , TI 1301-10 , 700 C

654 ; 66 , TI 1301-11 , 785 C

654 ; 67 , TI 1301-19 , 785 C

455 ; 68 , TI 1301-38 , 550 C

498 ; 69 , TI 1301-39 , 600 C

583 ; 70 , TI 1301-40 , 700 C

654 ; 71 , TI 1301-41 , 785 C

654 ; 72 , TI 1301-49 , 785 C

524 ; 73 , TR 1302-3 , 1000 C

163 ; 74 , TI 1301-6 , 200 C

200 ; 75 , AR 1301 , 2.0 VOL PCT

250 ; 76 , TR 1302-1 , 550 C

387 ; 77 , TR 1302-4 , 785 C

304 ; 78 , AR 1302 , 9.13 VOL PCT

625 ; 79 , TR 1302-6 , 1150 C

498 ; 80 , TR 1302-8 , 964 C

300 ; 81 , AR 1303 , 0.30 VOL PCT

550 ; 82 , TRC 1306 , 360 C

625 ; 83 , FRC 1301 , 13440 NM3/H

640 ; 84 , FRC 1302 , 18400 NM3/H

660 ; 85 , FR 1304 , 18680 KG/H

700 ; 86 , PRC 1301 , 35 KG/CM2G

567 ; 87 , FR 1305 , 11290 KG/H

500 ; 88 , TRC

603 ; 89 , FRC 1306 , 43300 KG/H

659 ; 90 , FRC 1308 , 8930 NM/H

250 ; 91 , TR 1302-2 , 550 C

640 ; 92 , PR 1306 , 32 KP/CM2G

455 ; 93 , TI 1301-8 , 550 C

498	:	94	,	TI	1301-9	,	600 C
583	:	95	,	TI	1301-10	,	700 C
654	:	96	,	TI	1301-11	,	785 C
654	:	97	,	TI	1301-19	,	785 C
455	:	98	,	TI	1301-38	,	550 C
498	:	99	,	TI	1301-39	,	600 C
583	:	100	,	TI	1301-40	,	700 C
654	:	101	,	TI	1301-41	,	785 C
654	:	102	,	TI	1301-49	,	785 C
524	:	103	,	TR	1302-3	,	1000 C
163	:	104	,	TI	1301-6	,	200 C
200	:	105	,	AR	1301	,	2.0 VOL PCT
250	:	106	,	TR	1302-1	,	550 C
387	:	107	,	TR	1302-4	,	785 C
304	:	108	,	AR	1302	,	9.13 VOL PCT
625	:	109	,	TR	1302-6	,	1150 C
498	:	110	,	TR	1302-8	,	964 C
300	:	111	,	AR	1303	,	0.30 VOL PCT
550	:	112	,	TRC	1306	,	360 C
654	:	113	,	FRC	1311	,	18600 KG/H
541	:	114	,	TI	1301-59	,	650 C
452	:	115	,	TI	1401-6	,	414 C
655	:	116	,	TRC	1403	,	231 C
260	:	117	,	TI	1401-14	,	240 C
135	:	118	,	AR	1401	,	0.27 VOL PCT
189	:	119	,	TI	1401-17	,	176 C
95	:	120	,	TI	1401-19	,	90 C
760	:	121	,	PIC	1407	,	3.8 KG/CM2G
478	:	122	,	FI	1403	,	8300 KG/H
656	:	123	,	FI	1506	,	8300 KG/H
160	:	124	,	TR	1502-5	,	150 C
649	:	125	,	FIC	1502	,	145 M3/H
640	:	126	,	FIC	1503	,	440 CM3/H
89	:	127	,	TR	1502-1	,	85 C
100	:	128	,	AR	1501	,	0.10 VOL PCT
350	:	129	,	IRC	1505	,	320 C
376	:	130	,	TR	1502-3	,	345 C
50	:	131	,	AR	1502	,	5 P.P.M.
356	:	132	,	AR	1504	,	74.25 VOL PCT
638	:	133	,	PI	1518	,	25.5 KP/CM2G
33	:	134	,	TR	1502-4	,	32 C
710	:	135	,	FI	1504	,	49000 NM3/H
33	:	136	,	TI	1602-24	,	32 C
701	:	137	,	PR	1603	,	280.5 KP/CM2G
694	:	138	,	FI	1602	,	250 000 NM3/H
0	:	139	,	TI	1602-25	,	0 C
184	:	140	,	TR	1601-1	,	172 C
463	:	141	,	TR	1601-2	,	424 C

530	; 142 ,	TR 1601-3	, 484 C
654	; 143 ,	FRC 1311	, 18600 KG/H
541	; 144 ,	TI 1301-59	, 650 C
452	; 145 ,	TI 1401-6	, 414 C
655	; 146 ,	TRC 1403	, 231 C
260	; 147 ,	TI 1401-14	, 240 C
135	; 148 ,	AR 1401	, 0.27 VOL PCT
189	; 149 ,	TI 1401-17	, 176 C
95	; 150 ,	TI 1401-19	, 90 C
760	; 151 ,	PIC 1407	, 3.8 KG/CM2G
478	; 152 ,	FI 1403	, 8300 KG/H
656	; 153 ,	FI 1506	, 8300 KG/H
160	; 154 ,	TR 1502-5	, 150 C
649	; 155 ,	FIC 1502	, 145 M3/H
640	; 156 ,	FIC 1503	, 440 CM3/H
89	; 157 ,	TR 1502-1	, 85 C
100	; 158 ,	AR 1501	, 0.10 VOL PCT
350	; 159 ,	IRC 1505	, 320 C
376	; 160 ,	TR 1502-3	, 345 C
50	; 161 ,	AR 1502	, 5 P.P.M.
356	; 162 ,	AR 1504	, 74.25 VOL PCT
638	; 163 ,	PI 1518	, 25.5 KP/CM2G
33	; 164 ,	TR 1502-4	, 32 C
710	; 165 ,	FI 1504	, 49000 NM3/H
33	; 166 ,	TI 1602-24	, 32 C
701	; 167 ,	PR 1603	, 280.5 KP/CM2G
694	; 168 ,	FI 1602	, 250 000 NM3/H
0	; 169 ,	TI 1602-25	, 0 C
184	; 170 ,	TR 1601-1	, 172 C
463	; 171 ,	TR 1601-2	, 424 C
530	; 172 ,	TR 1601-3	, 484 C
654	; 173 ,	FRC 1311	, 18600 KG/H
541	; 174 ,	TI 1301-59	, 650 C
452	; 175 ,	TI 1401-6	, 414 C
655	; 176 ,	TRC 1403	, 231 C
260	; 177 ,	TI 1401-14	, 240 C
135	; 178 ,	AR 1401	, 0.27 VOL PCT
189	; 179 ,	TI 1401-17	, 176 C
95	; 180 ,	TI 1401-19	, 90 C
760	; 181 ,	PIC 1407	, 3.8 KG/CM2G
478	; 182 ,	FI 1403	, 8300 KG/H
656	; 183 ,	FI 1506	, 8300 KG/H
160	; 184 ,	TR 1502-5	, 150 C
649	; 185 ,	FIC 1502	, 145 M3/H
640	; 186 ,	FIC 1503	, 440 CM3/H
89	; 187 ,	TR 1502-1	, 85 C
100	; 188 ,	AR 1501	, 0.10 VOL PCT

350	:	189	,	IRC	1505	,	320 C
376	:	190	,	TR	1502-3	,	345 C
50	:	191	,	AR	1502	,	5 P.P.M.
356	:	192	,	AR	1504	,	74,25 VOL PCT
638	:	193	,	PI	1518	,	25,5 KP/CM2G
33	:	194	,	TR	1502-4	,	32 C
710	:	195	,	FI	1504	,	49000 NM3/H
33	:	196	,	TI	1602-24	,	32 C
701	:	197	,	PR	1603	,	280,5 KP/CM2G
694	:	198	,	FI	1602	,	250 000 NM3/H
0	:	199	,	TI	1602-25	,	0 C
184	:	200	,	TR	1601-1	,	172 C
463	:	201	,	TR	1601-2	,	424 C
530	:	202	,	TR	1601-3	,	484 C

549	:	203	,	TR	1601-4	,	501 C
493	:	204	,	TR	1601-5	,	451 C
463	:	205	,	TI	1602-1	,	424 C
530	:	206	,	TI	1602-2	,	484 C
530	:	207	,	TI	1602-3	,	484 C
549	:	208	,	TI	1602-4	,	501 C
549	:	209	,	TI	1602-5	,	501 C
549	:	210	,	TL	1602-6	,	501 C
549	:	211	,	TI	1602-7	,	501 C
549	:	212	,	TI	1602-8	,	501 C
529	:	213	,	TI	1602-9	,	483 C
529	:	214	,	TI	1602-10	,	483 C
529	:	215	,	TI	1602-11	,	483 C
493	:	216	,	TI	1602-12	,	457 C
493	:	217	,	TI	1602-13	,	451 C
379	:	218	,	TR	1601-6	,	347 C
721	:	219	,	FI	1601	,	22600 KG/H
17	:	220	,	TI	1602-29	,	17 C
390	:	221	,	FRC	1603	,	3123 NM3/H
634	:	222	,	FI	1604	,	796 NM3/H
676	:	223	,	FR	1605	,	20560 KG/H
166	:	224	,	FI	1606	,	6522 KG/H

549	:	225	,	TR	1601-4	,	501 C
493	:	226	,	TR	1601-5	,	451 C
463	:	227	,	TI	1602-1	,	424 C
530	:	228	,	TI	1602-2	,	484 C
530	:	229	,	TI	1602-3	,	484 C
549	:	230	,	TI	1602-4	,	501 C
549	:	231	,	TI	1602-5	,	501 C
549	:	232	,	TL	1602-6	,	501 C
549	:	233	,	TI	1602-7	,	501 C
549	:	234	,	TI	1602-8	,	501 C
529	:	235	,	TI	1602-9	,	483 C

529 ; 236 , TI 1602-10 , 483 C
529 ; 237 , TI 1602-11 , 483 C
493 ; 238 , TI 1602-12 , 457 C
493 ; 239 , TI 1602-13 , 451 C
379 ; 240 , TR 1601-6 , 347 C
721 ; 241 , FI 1601 , 22600 KG/H
17 ; 242 , TI 1602-29 , 17 C
390 ; 243 , FRC 1603 , 3123 NM3/H
634 ; 244 , FI 1604 , 796 NM3/H
676 ; 245 , FR 1605 , 20560 KG/H
166 ; 246 , FI 1606 , 6522 KG/H

549 ; 247 , TR 1601-4 , 501 C
493 ; 248 , TR 1601-5 , 451 C
463 ; 249 , TI 1602-1 , 424 C
530 ; 250 , TI 1602-2 , 484 C
530 ; 251 , TI 1602-3 , 484 C
549 ; 252 , TI 1602-4 , 501 C
549 ; 253 , TI 1602-5 , 501 C
549 ; 254 , TL 1602-6 , 501 C
549 ; 255 , TI 1602-7 , 501 C
549 ; 256 , TI 1602-8 , 501 C
529 ; 257 , TI 1602-9 , 483 C
529 ; 258 , TI 1602-10 , 483 C
529 ; 259 , TI 1602-11 , 483 C
493 ; 260 , TI 1602-12 , 457 C
493 ; 261 , TI 1602-13 , 451 C
379 ; 262 , TR 1601-6 , 347 C
721 ; 263 , FI 1601 , 22600 KG/H
17 ; 264 , TI 1602-29 , 17 C
390 ; 265 , FRC 1603 , 3123 NM3/H
634 ; 266 , FI 1604 , 796 NM3/H
676 ; 267 , FR 1605 , 20560 KG/H
166 ; 268 , FI 1606 , 6522 KG/H

640 ; 269 , PRC 1701 , 16 KP/CM2G
500 ; 270 , PRC 1705 , 8 KP/CM2G
48 ; 271 , TI 1701-7 , 47 C
563 ; 272 , FI 1705 , 12000 NCM3/H
787 ; 273 , FRC 1702, R1 , 6000 KG/H
507 ; 274 , FRC 1702, R2 , 1100 KG/H
560 ; 275 , PRC 1710 , 14 KP/CM2G
629 ; 276 , FI 1706 , 2380 KG/H
575 ; 277 , FI 1709 , 8340 KG/H
375 ; 278 , FI 1801, R1 , 14700 KG/H
640 ; 279 , FI 1801, R2 , 4800 KG/H
640 ; 280 , FI 1801, R3 , 1200 KG/H
700 ; 281 , IRC 1805 , 70 C
500 ; 282 , LI 1901 A , 1349 M3

500	; 283 ,	LI 1901 B	, 1349 M3
500	; 284 ,	LI 1901 C	, 1349 M3
160	; 285 ,	TI 4801-7	, 150 C
661	; 286 ,	FI 4821	, 36600 KG/H
0	; 287 ,	KG 4821	, 292.8×10 KCAL/H
664	; 288 ,	FI 4822	, 53000 KG/H
0	; 289 ,	KC 4822	, 424×10 KCAL/H
73	; 290 ,	TI 4801-9	, 70 C
73	; 291 ,	TI 4801-10	, 70 C
386	; 292 ,	FI 4301	, 141.5 CM3/H
386	; 293 ,	FI 4401	, 141.5 CM3/H
666	; 294 ,	FI 4501	, 57100 KG/H
0	; 295 ,	DUMMY	
0	; 296 ,	DUMMY	
0	; 297 ,	DUMMY	
0	; 298 ,	DUMMY	
0	; 299 ,	DUMMY	
0	; 300 ,	DUMMY	
400	; 301 ,	PRC 2106	, 4.0 KP/CM2G
73	; 302 ,	TD 2101	, 70 C
671	; 303 ,	FRRC2101	, 10650 NM3/H
874	; 304 ,	FI 2101	, 113600 NM3/H
133	; 305 ,	TD 2102	, 125 C
437	; 306 ,	TD 2104	, 865 C
437	; 307 ,	TD 2106	, 865 C
437	; 308 ,	TD 2108	, 865 C
216	; 309 ,	TD 2105	, 200 C
216	; 310 ,	TD 2107	, 200 C
216	; 311 ,	TD 2109	, 200 C
788	; 312 ,	PRC 2152	, 31.5 KP/CM2G
344	; 313 ,	TD 2103	, 316 C
774	; 314 ,	FR 2102	, 44 T/H
563	; 315 ,	FR 2103	, 45 T/H
776	; 316 ,	FR 2105	, 18.5 T/H
600	; 317 ,	PDI 2118	, 3000 MM WG
881	; 318 ,	FRC 2109	, 13.21 M3/H
866	; 319 ,	FRC 2111	, 17320 NM3/H
600	; 320 ,	DIR 2115	, 1320 KG/CM3
41	; 321 ,	TD 2110	, 40 C
0	; 322 ,	WTC 2112	, 561
852	; 323 ,	FRC 2112	, 28.69 T/H
250	; 324 ,	FI 2108	, 125 L/H
742	; 325 ,	FR 2110	, 11200CM3/H
743	; 326 ,	FRA 2117	, 6036 CM3/H
250	; 327 ,	PD 2102	, 2.5 KP/CM2G
249	; 328 ,	FD 2111	, 4240 KG/H

400	; 329 ,	PRC 2106	, 4.0 KP/CM2G
73	; 330 ,	TD 2101	, 70 C
671	; 331 ,	FRR2101	, 10650 NM3/H
874	; 332 ,	FI 2101	, 113600 NM3/H
133	; 333 ,	TD 2102	, 125 C
437	; 334 ,	TD 2104	, 865 C
437	; 335 ,	TD 2106	, 865 C
437	; 336 ,	TD 2108	, 865 C
216	; 337 ,	TD 2105	, 200 C
216	; 338 ,	TD 2107	, 200 C
216	; 339 ,	TD 2109	, 200 C
788	; 340 ,	PRC 2152	, 31.5 KP/CM2G
344	; 341 ,	TD 2103	, 316 C
774	; 342 ,	FR 2102	, 44 T/H
563	; 343 ,	FR 2103	, 45 T/H
776	; 344 ,	FR 2105	, 18.5 T/H
600	; 345 ,	PDI 2118	, 3000 MM WG
881	; 346 ,	FRC 2109	, 13.21 M3/H
866	; 347 ,	FRC 2111	, 17320 NM3/H
600	; 348 ,	DIR 2115	, 1320 KG/CM3
41	; 349 ,	TD 2110	, 40 C
0	; 350 ,	WTC 2112	, 561
852	; 351 ,	FRC 2112	, 28.69 T/H

400	; 352 ,	PRC 2106	, 4.0 KP/CM2G
73	; 353 ,	TD 2101	, 70 C
671	; 354 ,	FRR2101	, 10650 NM3/H
874	; 355 ,	FI 2101	, 113600 NM3/H
133	; 356 ,	TD 2102	, 125 C
437	; 357 ,	TD 2104	, 865 C
437	; 358 ,	TD 2106	, 865 C
437	; 359 ,	TD 2108	, 865 C
216	; 360 ,	TD 2105	, 200 C
216	; 361 ,	TD 2107	, 200 C
216	; 362 ,	TD 2109	, 200 C
788	; 363 ,	PRC 2152	, 31.5 KP/CM2G
344	; 364 ,	TD 2103	, 316 C
774	; 365 ,	FR 2102	, 44 T/H
563	; 366 ,	FR 2103	, 45 T/H
776	; 367 ,	FR 2105	, 18.5 T/H
600	; 368 ,	PDI 2118	, 3000 MM WG
881	; 369 ,	FRC 2109	, 13.21 M3/H
866	; 370 ,	FRC 2111	, 17320 NM3/H
600	; 371 ,	DIR 2115	, 1320 KG/CM3
41	; 372 ,	TD 2110	, 40 C
0	; 373 ,	WTC 2112	, 561
852	; 374 ,	FRC 2112	, 28.69 T/H

400	; 375 ,	PRC 2106	, 4.0 KP/CM2G
-----	---------	----------	---------------

73	;	376	,	TD	2101	,	70 C
671	;	377	,	FRR	C2101	,	10650 NM3/H
874	;	378	,	FI	2101	,	113600 NM3/H
133	;	379	,	TD	2102	,	125 C
437	;	380	,	TD	2104	,	865 C
437	;	381	,	TD	2106	,	865 C
437	;	382	,	TD	2108	,	865 C
216	;	383	,	TD	2105	,	200 C
216	;	384	,	TD	2107	,	200 C
216	;	385	,	TD	2109	,	200 C
788	;	386	,	PRC	2152	,	31.5 KP/CM2G
344	;	387	,	TD	2103	,	316 C
774	;	388	,	FR	2102	,	44 T/H
563	;	389	,	FR	2103	,	45 T/H
776	;	390	,	FR	2105	,	18.5 T/H
600	;	391	,	PDI	2118	,	3000 MM WG
881	;	392	,	FRC	2109	,	13.21 M3/H
866	;	393	,	FRC	2111	,	17320 NM3/H
600	;	394	,	DIR	2115	,	1320 KG/CM3
41	;	395	,	TD	2110	,	40 C
0	;	396	,	WTC	2112	,	561
852	;	397	,	FRC	2112	,	28.69 T/H

600	;	398	,	PRC	3307	,	6 KP/CM2G
467	;	399	,	TRC			
802	;	400	,	FRTC	3101	,	5050 KG/H
802	;	401	,	FRTC	3102	,	5050 KG/H
530	;	402	,	PRC	3304	,	3.3 KP/CM2G
804	;	403	,	FRC	3103	,	18.96 T/H
804	;	404	,	FRC	3104	,	18.96 T/H
700	;	405	,	TRC	3307	,	16 C
720	;	406	,	TRC	3110	,	180 C
720	;	407	,	TRC	3111	,	180 C
617	;	408	,	PRC	3108	,	3.7 KP/CM2G
617	;	409	,	PRC	3109	,	3.7 KP/CM2G
800	;	410	,	FRC	3105	,	1800 KG/H
800	;	411	,	FRC	3106	,	1800 KG/H
250	;	412	,	PRC	3113	,	-0.25 KP/CM2G
250	;	413	,	PRC	3114	,	-0.25 KP/CM2G
250	;	414	,	PRC	3115	,	0.25 KP/CM2A
250	;	415	,	PRC	3116	,	0.25 KP/CM2A
725	;	416	,	TR	3124	,	145 C
800	;	417	,	FRT	3228	,	7600 KG/H
357	;	418	,	TR	3234	,	145 C
357	;	419	,	TR	3235	,	145 C
357	;	420	,	TR	3236	,	145 C
357	;	421	,	TR	3237	,	145 C
436	;	422	,	DR	3001	,	1320 KG=CM3

0	:	423	,	DUMMY	
750	:	424	,	PRC 3044	, 12 KP/CM2G
760	:	425	,	TRC 3070	, 190 C
250	:	426	,	PRC 3046	, 4 KP/CM2G
600	:	427	,	PRC 3307	, 6 KP/CM2G
467	:	428	,	TRC	
802	:	429	,	FRTC3101	, 5050 KG/H
802	:	430	,	FRTC3102	, 5050 KG/H
530	:	431	,	PRC 3304	, 3.3 KP/CM2G
804	:	432	,	FRC 3103	, 18.96 T/H
804	:	433	,	FRC 3104	, 18.96 T/H
700	:	434	,	TRC 3307	, 16 C
720	:	435	,	TRC 3110	, 180 C
720	:	436	,	TRC 3111	, 180 C
617	:	437	,	PRC 3108	, 3.7 KP/CM2G
617	:	438	,	PRC 3109	, 3.7 KP/CM2G
800	:	439	,	FRC 3105	, 1800 KG/H
800	:	440	,	FRC 3106	, 1800 KG/H
250	:	441	,	PRC 3113	, -0.25 KP/CM2G
250	:	442	,	PRC 3114	, -0.25 KP/CM2G
250	:	443	,	PRC 3115	, 0.25 KP/CM2A
250	:	444	,	PRC 3116	, 0.25 KP/CM2A
725	:	445	,	TR 3124	, 145 C
800	:	446	,	FRT 3228	, 7600 KG/H
357	:	447	,	TR 3234	, 145 C
357	:	448	,	TR 3235	, 145 C
357	:	449	,	TR 3236	, 145 C
357	:	450	,	TR 3237	, 145 C
600	:	451	,	PRC 3307	, 6 KP/CM2G
467	:	452	,	TRC	
802	:	453	,	FRTC3101	, 5050 KG/H
802	:	454	,	FRTC3102	, 5050 KG/H
530	:	455	,	PRC 3304	, 3.3 KP/CM2G
804	:	456	,	FRC 3103	, 18.96 T/H
804	:	457	,	FRC 3104	, 18.96 T/H
700	:	458	,	TRC 3307	, 16 C
720	:	459	,	TRC 3110	, 180 C
720	:	460	,	TRC 3111	, 180 C
617	:	461	,	PRC 3108	, 3.7 KP/CM2G
617	:	462	,	PRC 3109	, 3.7 KP/CM2G
800	:	463	,	FRC 3105	, 1800 KG/H
800	:	464	,	FRC 3106	, 1800 KG/H
250	:	465	,	PRC 3113	, -0.25 KP/CM2G
250	:	466	,	PRC 3114	, -0.25 KP/CM2G
250	:	467	,	PRC 3115	, 0.25 KP/CM2A
250	:	468	,	PRC 3116	, 0.25 KP/CM2A
725	:	469	,	TR 3124	, 145 C

800	; 470 ,	FRT 3228	, 7600 KG/H
357	; 471 ,	TR 3234	, 145 C
357	; 472 ,	TR 3235	, 145 C
357	; 473 ,	TR 3236	, 145 C
357	; 474 ,	TR 3237	, 145 C
791	; 475 ,	FRTC3212	, 11.67 CM3/H
791	; 476 ,	FRTC3213	, 11.87 CM3/H
791	; 477 ,	FRTC3214	, 11.87 CM3/H
791	; 478 ,	FRTC3215	, 11.87 CM3/H
411	; 479 ,	TR 3238	, 170 C
411	; 480 ,	TR 3239	, 170 C
411	; 481 ,	TR C240	, 170 C
411	; 482 ,	TR 3241	, 170 C
444	; 483 ,	FR 3216	, 5000 NM3/H
444	; 484 ,	FR 3217	, 5000 NM3/H
444	; 485 ,	FR 3218	, 5000 NM3/H
444	; 486 ,	FR 3219	, 5000 NM3/H
367	; 487 ,	TR 3242	, 150 C
367	; 488 ,	TR 3243	, 150 C
367	; 489 ,	TR 3244	, 150 C
367	; 490 ,	TR 3245	, 150 C
900	; 491 ,	TRC 3226	, 180 C
900	; 492 ,	TRC 3227	, 180 C
900	; 493 ,	TRC 3228	, 180 C
900	; 494 ,	TRC 3229	, 180 C
300	; 495 ,	TR 3355	, 120 C
254	; 496 ,	TR 3356	, 100 C
75	; 497 ,	TR 3351	, 28 C
791	; 498 ,	FRTC3212	, 11.67 CM3/H
791	; 499 ,	FRTC3213	, 11.87 CM3/H
791	; 500 ,	FRTC3214	, 11.87 CM3/H
791	; 501 ,	FRTC3215	, 11.87 CM3/H
411	; 502 ,	TR 3238	, 170 C
411	; 503 ,	TR 3239	, 170 C
411	; 504 ,	TR C240	, 170 C
411	; 505 ,	TR 3241	, 170 C
444	; 506 ,	FR 3216	, 5000 NM3/H
444	; 507 ,	FR 3217	, 5000 NM3/H
444	; 508 ,	FR 3218	, 5000 NM3/H
444	; 509 ,	FR 3219	, 5000 NM3/H
367	; 510 ,	TR 3242	, 150 C
367	; 511 ,	TR 3243	, 150 C
367	; 512 ,	TR 3244	, 150 C
367	; 513 ,	TR 3245	, 150 C
900	; 514 ,	TRC 3226	, 180 C
900	; 515 ,	TRC 3227	, 180 C
900	; 516 ,	TRC 3228	, 180 C

900	; 517	, TRC 3229	, 180 C
300	; 518	, TR 3355	, 120 C
254	; 519	, TR 3356	, 100 C
75	; 520	, TR 3351	, 28 C
791	; 521	, FRTC3212	, 11.67 CM3/H
791	; 522	, FRTC3213	, 11.87 CM3/H
791	; 523	, FRTC3214	, 11.87 CM3/H
791	; 524	, FRTC3215	, 11.87 CM3/H
411	; 525	, TR 3238	, 170 C
411	; 526	, TR 3239	, 170 C
411	; 527	, TR C240	, 170 C
411	; 528	, TR 3241	, 170 C
444	; 529	, FR 3216	, 5000 NM3/H
444	; 530	, FR 3217	, 5000 NM3/H
444	; 531	, FR 3218	, 5000 NM3/H
444	; 532	, FR 3219	, 5000 NM3/H
367	; 533	, TR 3242	, 150 C
367	; 534	, TR 3243	, 150 C
367	; 535	, TR 3244	, 150 C
367	; 536	, TR 3245	, 150 C
900	; 537	, TRC 3226	, 180 C
900	; 538	, TRC 3227	, 180 C
900	; 539	, TRC 3228	, 180 C
900	; 540	, TRC 3229	, 180 C
300	; 541	, TR 3355	, 120 C
254	; 542	, TR 3356	, 100 C
75	; 543	, TR 3351	, 28 C
0	; 544	CHECK TEMP	
500	; 545	, -	0-20 MV
500	; 546	, -	0-50 MV
500	; 547	, -	0-1 V
500	; 548	, -	0-10 V
0	; 549	, -	TEMP

```
W H1:  AL. W0      H0. ; TAIL SEGMENT 9:
        RS W0      B18 ; INITIALIZE(SIMULATION TABLE);
        AL. W0      H1. ;
        AL W1       0   ;
        AL W2      A54>14^1 ;
        JL W3      (B39) ; CALL LOADER;

        JL.        H1. ;
```

```
E
E
L ; END S9
N
T
```


C [A55 >13 ^ 1] - 1

S C10, E20, F20, G16, H5, I4

W K=0

; SEGMENT 10:

; SELF CHECK:

; THE PROGRAM IS INTENDED TO CHECK THE
 ; HARDWARE OF THE CENTRAL PROCESSOR. MOST
 ; LIKELY HARDWARE ERRORS WILL CAUSE THE NORMAL
 ; PROGRAM EXECUTION TO BE DISTURBED TO SUCH AN
 ; EXTENT THAT THE SELF CHECK PROGRAM WILL NOT
 ; BE EXECUTED.

; THE PROGRAM MAKES A SYSTEMATIC TEST
 ; OF MICRO-COMMANDS AND MICRO-JUMP
 ; CONDITIONS. (EXCLUSIVE ARE COMMANDS AND JUMP
 ; CONDITION WHICH CONCERNS THE INTERRUPT SYSTEM
 ; AND THE INPUT/OUTPUT FUNCTIONS).

; MESSAGES FOR HARDWARE ERRORS DETECTED BY
 ; THE PROGRAM ARE PRINTED ON THE CONTROL TYPE -
 ; WRITER. AN ADDRESS WILL INDICATE THE POINT
 ; IN THE PROGRAM WHERE THE ERROR HAS BEEN
 ; DETECTED.

; THE PROGRAM WILL PERIODICALLY PRINT A
 ; CONTROL MESSAGE ON THE CONTROL TYPEWRITER.
 ; ABSENCE OF THIS MESSAGE WILL INDICATE
 ; THAT A HARDWARE ERROR HAS PREVENTED
 ; THE PROGRAM FROM BEING EXECUTED.

; THE PERIOD FOR THE CONTROL MESSAGE CAN
 ; BE SPECIFIED BY THE PERIOD OPTION FOR
 ; TASK SELF-CHECK.

; NORMALLY, THE PROGRAM IS EXECUTED
 ; CONTINUOUSLY, BUT IT MAY BE STOPPED BY
 ; THE CLOSE OPTION.

; THE CHECK PERFORMED BY THE DIFFERENT
 ; PART OF THE PROGRAM IS DOCUMENTED IN
 ; THE COMMENTS.

H I1: <HHMM SELF-CHECK <10>> ; STRING

I2: <OK <63><10>>

I3: <ALARM <0>>

I4: <DDDD <63><10>>

W

H G0: 000001 000001 , G1: 000001 000001 ; CONSTANT

G2: -1 , G3: -1

G4: 1 < 10 , G5: 0

G6: 0 , G7: 0

W G8: 000000 000000 111111 011011
 G10: 100100 000000 111111 100000
 G12: 1 < 23
 G14: 000000 000000 111110 111111
 G16: 6

H H0: 0 , H1: 0 ; VARIABLE
 H2: 0 , H3: 0 ;
 H4: 0 , H5: 0 ;

; PROCEDURE CHECK MES;
 ; W2: LINK/DESTROYED;
 ; W0, W1, W3 DESTROYED;

W C1: RS. W2 H0. ; BEGIN
 RL W1 B38 ;
 JL W3 (B19) ; I1:= CON CL (CLOCK);
 0 ;
 RS. W1 I1. ;
 AL. W0 I1. ;
 JL W3 (B26) ; OUT AN (I1);
 JL. (H0.) ; END;

C2: JL. W2 C1. ; EXIT: CHECK MES;
 AL. W0 I2. ;
 JL W3 (B26) ; OUT AN (I2);
 H A50 , A32 ; EXIT TASK;

W C3: AC. W0 I1. ;
 WA W0 4 ; ALARM: COMMENT W2 = LINK;
 JL W3 (B24) ; I4:= CON AN (LINK);
 H I4. , 4 ;
 W JL. W2 C1. ; CHECK MES;
 AL. W0 I3. ;
 JL W3 (B26) ; OUT AN (I3);

C4: RL W0 A48+16 ; START:
 SE. W0 (G4.) ; IF NEXT EX [TASK 9] = INFINITE
 SH W0 (B 38) ; ~ NEXT EX [TASK 9] < CLOCK
 JL. C2. ; THEN GOTO EXIT;
 H A50 , A37 ; PASS CONTROL;

; PART A: TRANSPORT BETWEEN BUS AND W-REGISTERS.

; MC (32): BUS:= W[FR(10, 11)];
 ; MC (40): BUS:= W[PRE];
 ; MC (49): W[PRE]:= BUS;


```

; MC (55): BUS:= WCFR(6, 7)];
; MC (64): WCFR(6, 7)] (0:11):= BUS (0:11)];
; MC (65): WCFR(6, 7)] (12:23):= BUS(12:23)];

```

```

W C5: AL W0 0 ; A0: W0:= 0; [CALL (0)]
F1: LD W1 -24 ; A1: W1:= W0;
LD W2 X1 -24 ; A2: W2:= W1;
LD W3 X2 -24 ; A3: W3:= W2;
LD W0 X3 -24 ; A4: W0:= W3;
LD W0 24 ; A5: W3:= W0;
LD W3 24 ; A6: W2:= W3;
LD W2 24 ; A7: W1:= W2;
LD W1 24 ; A8: W0:= W1;
SZ W0 -1 ; A9: IF W0 = 0 THEN
E1: JL W2 C3. ; ALARM (1);

AL W0 -1 ; A10: W0:= -1; [CALL (1)]
LD W1 -24 ; A11: W1:= W0;
LD W2 -24 ; A12: W2:= W1;
LD W3 -24 ; A13: W3:= W2;
LD W0 -24 ; A14: W0:= W3;
LD W0 24 ; A15: W3:= W0;
F2: LD W3 X3 +25 ; A16: W2:= W3;
LD W2 X2 +25 ; A17: W1:= W2;
LD W1 X1 +25 ; A18: W0:= W1;
SO W0 -1 ; A19: IF W0 = -1 THEN
E2: JL W2 C3. ; ALARM (2);

```

; OTHER COMMANDS TESTED:

```

;A0:A10: MC(19): SB(0:11):= 12 EXB BUS(12); [CALL (0), ALL (1)]
;A9:A19: MC(27): BUS:= AND; [SB=ALL (1) ^ AR = ALL(0), SB = ALL (0) ^ AR=
;A1:A11: MC(38): BUS:= AR(0:23); [CALL (0), ALL (1)]
;A5:A15: MC(39): BUS:= BR; [CALL (0), ALL (1)]
;A1:A11: MC(53): BR:= BUS; [CALL (0), ALL (1)]
;A0:A10: MC(59): BUS:= SB; [CALL (0), ALL (1)]
;A9:A19: MC(60): SB(12:23):= BUS(12:23) [CALL (0), ALL (1)]
;A5:A15: MC(66,58,26x4): AR:= BUS(0) CON BUS; [CALL (0), ALL (1)]
;A9:A19: MC(67,68,69,70x1): SB(0:11):= BUS(0:11); [CALL (0), ALL (1)]
;A5:A15: MC(67,68,69,70x4): LSHL AR CON BR; [CALL (0), ALL (1)]
;A1:A11: MC(67,68,69,70x6): LSHR AR CON BR; [CALL (0), ALL (1)]
;A1: MC(67,68,69,70x8): STRATUS:= 1;
;A5: MC(67,68,69,70x9): STRATUS:= 0;

```

; JUMP CONDITION TESTED:

```

;A1:A5: SB(0):
;A1:A5: STRATUS:
;A1:A2:A3:A4: FR(10, 11):

```

; PART B: TEST SHIFT.

```

W C6:  RL. W0          G8.      ; B0:  W0:= 000000 000000 111111 011011;
        RL. W1          G10.    ; B1:  W1:= 100100 000000 111111 100000;
        AD  W1          6        ; B2:  W0 CON W1:= W0 CON W1 LSHL 6;
F3:    SX              10       ; B3:  IF EX (22) THEN
E3:    JL. W2          C3.      ;      ALARM (3);

        AD  W1          6        ; B4:  A: W0 CON W1:= W0 CON W1 LSHL 6;
F4:    SX              10       ; B5:  IF EX (22) THEN
        JL.            -4        ;      GOTO A;

        SE. W0          (G12.)   ; B7:  IF W0 CON W1 = 1 < 47 THEN
E4:    JL. W2          C3.      ;      ALARM (4);

        AD  W1          -47      ; B8:  W0 CON W1 := W0 CON W1 ASHR 47;
F5:    SE  W1          -1        ; B9:  IF W0 CON W1 = -1 THEN
E5:    JL. W2          C3.      ;      ALARM (5);

        BZ  W1          1        ; B10: W1 (0:11):= 0;
F6:    AD  W2          -35      ; B11: W1 CON W2:= W1 CON W2 ASHR 35;
        SN  W1          0        ; B12:
        SE  W2          1        ; B13: IF W1 CON W2 = 1 THEN
E6:    JL. W2          C3.      ;      ALARM (6);

```

; COMMANDS TESTED:

```

;B2:B4: MC(41): TEST SHIFT: EX(22):= AR(-1) EXOR AR(0) ~ EX(22)
;      [ALL 8 COMBINATIONS]
;B2:B8 :A1: MC(51): SC:= BUS(17:23);      [ALL (0), ALL (1)]
;B8:    MC(54): SC:= SC - 1;
;B8:B11: MC(67,68,69,70x7): ASHR AR CON BR; [AR(0)=0, AR(0)=1]

```

; JUMP CONDITION TESTED:

```

;B8:    SC = 0;
;B10:A19: FR(0,1,2,3,4,5) -, MODIF:
;B7: A1:  FR(8,9):
;B0: B7: A2: FR(0,1,2,3,4,5):

```

; PART C: TEST COMBINED JUMP CONDITION.

; IF SB = 0: IF AR = 0: IF AR < 0:

```

B D3          ; COUNT:= W1 = 0; BIT:= W2 = 1;
W C7:
D1:  AL  W0          1        ; C1: SET TEST:

```

```

      LS  W0      X2      ; C2:      TEST (IF SB = 0);
      SE  W2      0      ; C3:      TEST (IF AR = 0);
F7:   SH  W2      0      ; C4:      TEST (IF AR < 0);
      JL.         D3.    ; C5:      IF COUNT = 23 x COUNT = 24 THEN
D2:   SN  W0      1      ; C6: TEST:
E7:   JL. W2      C3.    ;         IF W0 = 1 THEN ALARM(7);
      LS  W2      1      ; C7:      BIT:= BIT SHIFT 1;
      AL  W1      X1  +1  ; C8:      COUNT:= COUNT + 1;
      JL.         D1.    ; C9:      GOTO SET TEST

      D3:   SN  W1      23   ; C10: END:
      JL.         D2.    ; C11:     IF COUNT = 23 THEN GOTO TEST;
F8:F9: SL  W0      1      ; C12:
      SL  W0      2      ; C13:     IF W0 = 1 THEN
E8:   JL. W2      C3.    ;         ALARM(8);
      SH  W2      0      ; C14:     TEST (IF AR < 0);
      SE  W1      24   ; C15:     IF COUNT = 24 THEN
E9:   JL. W2      C3.    ;         ALARM(9);
E

```

; JUMP CONDITION TESTED:

```

;C2:      SB = 0
;C12:C13: AR(-1)
;C3:      AR = 0
;C4: C14: AR < 0

```

; PART D: HALFWORD, NORMALIZE, AND TEST INTEGER.

```

W C8:  HL. W0      G2.    ; D0:
      HS  W0      0      ; D1:  W0:= -1;
F11:   NS. W0      H1.    ; D2:  H1:= -23; W0:= 1 SHIFT 23;
      NS. W0      H2.    ; D3:  H2:= 0;

      BZ. W0      G6.    ; D4:  W0:= 0;
      NS. W0      H4.    ; D5:  H4:= -2048

      AL  W1      16     ; D6:  W0 CON W1:= 1 SHIFT 4
      ND. W1      H3.    ; D7:  H3:= -42; W0:= 1 SHIFT 22;
      NS. W0      H5.    ; D8:  H5:= 0;

      WA. W0      G4.    ; D9:  W0:= W0 + 1 SHIFT 22;
      SX         2      ; D10: TEST INTEGER (0, 1);
      WA. W0      H4.    ; D11: W0:= W0 + H4 CON H5;
      SX         2      ; D12: TEST INTEGER(1, 0);
      BA. W0      H1.    ; D13: W0:= W0 + H1;

```

```

F10: SX                2          ; D14: TEST INTEGER(1, 1);
E10: JL. W2            C3.        ;      IF EX (22) THEN ALARM (10);
      WA. W0            H2.        ; D15: W0:= W0 + H2 CON H3;
      SN. W0            (G14.)     ; D16: TEST INTEGER(0, 0);
      SX                2          ; D17: IF W0 = 0 CON (-65) v EX(22)
E11: JL. W2            C3.        ;      THEN ALARM(11);

```

; BIT PATTERN FOR THE SUMMATION:

```

;W0:      010000 000000          000000 000000
;G4:      010000 000000          000000 000000
;SUM:     0100000 000000          000000 000000
;H4:      100000 000000 , H5:    000000 000000
;SUM:     1000000 000000          000000 000000
;      111111 111111 , H1:      111111 101001
;SUM:     1111111 111111          111111 101001
;H2:      000000 000000 , H3:    111111 010110
;SUM:     0000000 000000          111110 111111

```

; COMMANDS TESTED:

```

; D5          MC(28,29x1):        BUS:= -2048;
; D2, D7      MC(36):            BUS:= 12 EXIT 0 CON 5 EXIT (17)
;                                     (17:23);
; D0, D4      MC(37):            BUS:= 12 EXIT 0 CON SB(0:11);
; D1, D3      MC(67,68,69,70x2):  SB(0:11):= BUS(12:23);
; D9, D11, D13, D15 MC(67,68,69,70x14): TEST INTEGER;

```

; JUMP CONDITION TESTED:

```

; D2, D3, D7, D8: AR(0) = AR(1);
; D2, D7:        AR(1) = AR(2);

```

```

; PART E: JUMP, DIVIDE, MULTIPLY, AND- COMMAND,
;          AND EXEPTION REGISTER.

```

B D2

```

W C9:  AL  W0          -1          ; E0:
      AL  W1          -76         ; E1:  W0 CON W1:= -76;
      JL.           2           ; E2:
      WD. W1          G16.        ; E3:  DIVIDE (W0 CON W1) BY: (6);
F12:  SN  W0          -4          ; E4:  IF REM = -4
      SE  W1          -12         ; E5:  v QUOTIENT = -12 THEN
E12:  JL. W2          C3.         ;      ALARM(12);
      JL. W1          D1.         ; E6:  GOTO A;

```

```

      JL  W3      X2      ; E7: B: GOTO C;
      AL  W1      -15     ; E8: D:
      WM. W1      G2.     ; E9: W0 CON W1:= -15 x (-1);
F13: JL.          D2.     ; E10: GOTO E;

D1:  JL  W2      X1      ; E11: A: GOTO B;
      JL          (6)     ; E12: C: GOTO D;

D2:  SN  W0      0        ; E13: E: IF W0 CON W1 = 15 THEN
      SE  W1      15      ; E14: ALARM(13);
E13: JL. W2      C3.     ;

      AL  W1      -1      ; E15:
F14: WM. W1      G6.     ; E16: W0 CON W1:= -1 x 0
      SN  W0      0        ; E17:
      SE  W1      0        ; E18: IF W0 CON W1 = 0 THEN
E14: JL. W2      C3.     ; ALARM(14);

      LX. W0      G6.     ; E19: W0:= 0 EXOR 0;
F15: SE  W0      0        ; E20: IF W0 = 0 THEN
E15: JL. W2      C3.     ; ALARM(15);
      AL  W0      -1      ; E21:
      LX. W0      G6.     ; E22: W0:= -1 EXOR 0
F16: SE  W0      -1      ; E23: IF W0 = -1 THEN
E16: JL. W2      C3.     ; ALARM(16);

      XL.          G6.     ; E24:
F17: XS          0        ; E25: W0(0:11):= EX:= 0;
      XL.          G2.     ; E26:
      XS          1        ; E27: W0(12:23):= EX:= 3;
      SE  W0      3        ; E28: IF W0 = 3 THEN
E17: JL. W2      C3.     ; ALARM(17);

```

E

; COMMOD TESTED:

```

; E9, E16:      MC(26): BUS:= SUM(-1:22);
; E19, E22:      MC(27): BUS:= AND
;                [SB=ALL(0) ^ AR=ALL(0), SB=ALL(1) ^ AR=ALL(1)]
; E3:           MC(28,29x2): BUS:= 24
; E25, E27:      MC(33): BUS: 22 EXT 0 CON EX;
; E2, E6, E10, D11: MC(47): FR:= BUS(0:11);
; E24, E26:      MC(48): EX:= BUS(22, 23);
; E3:           MC(52): DIVSUB:= IF AR(-1) = SB(0) THEN 1 ELSE 0,
; E9, E16:      MC(66,58,26x5):
;                AR CON BR:= SUM(-1) CON SUM(-1:22) CON SUM(23
;                BR(0:22)
; E3:           MC(67,68,69,70x5): LSHL BR;
; E3:           MC(67,68,69,70x15): BR(23):= 1

```

; JUMP CONDITION TESTED:

```
; E3          AR(-1) = SB(0)
; E9, E16     BR(22)
; E9, E16     BR(23)
; E3          BR(0) = BR(1)
; E6, E7, E11 JUMP LINK
```

; PART F: TEST OF ADDER.

; THE ADDER IS DIVICTED INTO 8 SECTIONS OF 3 BIT EACH,
; (DUE TO ITS CARRY LOOK AHEAD LOGIC).

; 1. FOR EACH 3 BIT SECTION THE POSSIBLE 8 VALUES OF THE SB-
; REGISTER IS ADDED TO THE POSSIBLE 8 VALUES OF THE AR REGISTER
; AND THE RESULT IS TESTED.

; 2. 1 IS EXECUTED BOTH WITHOUT AND WITH A CARRY COMING
; FROM THE PRECEEDING SECTION, FOR THE PURPOSE OF INTRODUCING
; A CARRY TO THE RIGHT MOST SECTION ALL ADDITION WITH
; CARRIES ARE EXECUTED AS A SUBTRACTION OF THE NEGATED
; VALUE IN SB.

; 3. IN EACH ADDITION A BIT PATTERN INCLUDING EVERY SECOND
; 3 BIT SECTION IS TESTED, THEREFORE 2 IS EXECUTED FIRST
; FOR THE BIT PATTERN RIGHT JUSTIFIED AND THEN LEFT
; JUSTIFIED.
; WHEN ADDING A SECTION, A CARRY MAY BE PRODUCED.
; IN CASE OF LEFT JUSTIFICATION AND PRODUCED CARRY
; THE CARRY FROM THE LEFTMOST SECTION IS
; REGISTERED IN THE EXCEPTION REGISTER

B D5

```
; W0:
; W1: A PRECALCULATED RESULT: R;
; W2: A BIT PATTERN FOR INCREMENTING: ONES;
; W3: BOOLEAN SHIFT, CARRY; INTEGER SBC, ARC;
```

; THE CONSTANT AND VARIABLE USED ARE:

```
; G0: PAT;
; G2: -1;
; H0: INTEGER SBR,
; H2: SB,
; H4: AR;
```

```

W C10: AL W3      0      ; SHIFT:= CARRY:= FALSE; SBC:= ARC:= 0;
                                ;
D1:  RL W2      G0.      ; OUTER LOOP:
    SZ W3      10 000 000 ; ONES:= IF SHIFT THEN PAT SHIFT 3 ELSE P
    LS W2      3        ;
    AL W1      0        ; R:= IF CARRY THEN ONES ELSE 0;
    SZ W3      01 000 000 ;
    AL W1      X2       ;

D2:  RS W1      H0.      ; SB LOOP: SBR:= R;
    AL W0      0        ;
    SZ W3      01 000 000 ;
    AL W0      X2       ;
    LS W0      -1       ; AR:= IF CARRY THEN ONES SHIFT(-1) ELSE
    RS W0      H2.      ;
    WA W0      2        ;
    SZ W3      01 000 000 ;
    WS W0      4        ;
    SZ W3      01 000 000 ; SB:= IF CARRY
    LX W0      G2.      ; THEN NEG (R + ONES SHIFT (-1))
    RX W0      H2.      ; ELSE R;
    SO W3      11 000 000 ;
    JL W0      4        ; IF SHIFT^CARRY THEN AR:=AR-1;
    BS W0      1        ;

D3:  RS W0      H4.      ; AR LOOP:
    SZ W3      01 000 000 ;
    WS W0      H2.      ; W0:= AR + IF CARRY THEN
    SO W3      01 000 000 ; (-SB) ELSE
F18: WA W0      H2.      ; (+SB);
    SE W0      X1       ; IF W0 = R
E18: JL W2      C3.      ; THEN ALARM(18);
    SO W3      10 000 000 ; IF SHIFT THEN
    JL W0      D5.      ; BEGIN
    SO W1      01 000 000 ; IF
    JL W0      D4.      ; R BIT 17 ^ -, EX(23);
F19: SX      1        ;
    JL W0      D5.      ; ^ -, R BIT 17 ^ EX(23)
E19: JL W2      C3.      ; THEN ALARM(19, 20);
F20:D4: SX      1        ;
E20: JL W2      C3.      ; END;

D5:  RL W0      H4.      ;
    WA W0      4        ; AR:= AR + ONES;
    WA W1      4        ; R:= R + ONES;
    AL W3      X3 +1     ; ARC:= ARC + 1;
    SZ W3      111      ; IF ARC < 8
    JL W0      D3.      ; THEN GOTO AR LOOP ELSE ARC:= 0;

```

```

RL. W1          H0.      ; SBC:= SBC + 1;
WA W1           4        ; R:= SBR + ONES;
SZ W3           111 000  ; IF SBC < 8
JL.             D2.      ; THEN GOTO SB LOOP ELSE SBC:= 0;

SO W3           100 000 000 ; IF -, CARRY v -, SHIFT THEN
JL.             D1.      ; BEGIN IF CARRY THEN SHIFT:= TRUE;
                  ; CARRY:= -, CARRY;
                  ; GOTO OUTER LOOP
E               ; END;

JL.             C4.      ; GOTO START;

```

```

B E1               ; TAIL SEGMENT 10:

W E0: RL. W0       E1.      ; TAIL:
      RS W0       A46+16 ; NAME TAB [TASK 8]:= <SELF>;
      AL W0       C5.      ;
      RS W0       A47+16 ; ADDR TAB [TASK 8]:= START;
      AL W0       0        ;
      RS W0       A48+16 ; NEXT EX [TASK 8]:= 0;
      AL W0       900      ;
      RS W0       A49+16 ; PERIODE [TASK 8]:= 15 MIN;
      AL W0       E0.      ;
      AL W1       0        ; CALL LOADER;
      AL W2       A54>13^1 ;
      JL W3       (B39)    ;

```

E1: <SELF>

```

JL.             E0.      ; ENTRY FROM LOADER: GOTO TAIL;

```

E ; END TAIL

L
; IDLIST S10
N T

E ; END SEGM 10
E ; END COND. SEGM 10

L
; END S10
N T


```
; SEGMENT 11:
;   THE SYSTEM BLOCK OF A, B AND E NAMES
;   IS CLOSED TO AVOID THE OVERLOADING OF THE SLANG
;   ASSEMBLER, THE FOLLOWING A AND B NAMES
;   ARE SAVED:
```

```
I0 = B28          ; H ENTRIES
I1 = B30          ; A/D SCAN TABLE
I2 = B31          ; - - - PART 2 REL
I3 = B32          ; ALARM SCAN TABLE
I4 = B33          ; PULSE SCAN TABLE
I5 = B15          ; - - - PART 2
I6 = [A55^1<12]<11~[A54^1<12]<10~[A55>15^1]
I7 = B39          ; LOADER
```

E

C
S

```
[I6>23^1]-1
A550, B32, E20, C11, J5, G13, K=0
```

```
; THE SCAN TABLE HAS ONE RECORD FOR EVERY A/D AND
; DIGITAL INPUT TERMINAL. THE RECORDS (OR ENTRIES)
; ARE REFERENCED BY THEIR ORDINAL NUMBER IN THE
; TABLE (CALLED SCAN NUMBER). HOWEVER WITHIN THE
; SYSTEM, THE ENTRIES CAN BE NAMED BY THEIR ADDRESS,
; RELATIVE TO THE TABLE BOTTOM. FURTHERMORE THERE
; ARE DIRECTIVE RECORDS, CALLED CR, GROUP
; OR END FOR CONTROL PURPOSES.
```

```
; THE FORMAT OF THE RECORDS:
```

```
; BITS IN A BYTE      1 1 1 1 1 1 1 1 1 1 1
; A/D ENTRY:          0 0 H L L I R E QREF ]
;                     [ HIGH LIMIT          ]
;                     [ LOW LIMIT           ]
;                     [ TERMINAL ADDRESS    ]
;                     [ FIRST PARAM         ]
;                     :
```

```
; DIGITAL INPUT:      0 0 0[BIT NO ] [WORD ]
; GROUP:              1 0 1[ALARM  ] [GROUP ]
; CR                  1 1 0 0 0 0 0 0 0 0 0
; END:                1 0 0 0 0 0 0 0 [END NO ]
```

```
; GROUP AND CR CAN APPEAR TOGETHER, AS
; THEIR LOGICAL SUM,
; THESE ITEMS ARE DESCRIBED IN THE TABLES
; BY MEANS OF SYMBOLIC NAMES, IN THE FOLLOWING
; MANNER:
```

```
J1 = 512          ; BIT HIGH
J2 = 256          ; - LOW
```

J3 = 128 ; - LOGGED
J4 = 64 ; - INTEGRATED
J5 = 32 ; - REMOVED

; EQREF: (E20 REFERS TO EQ 00)

E0 = 4 , E1 = 5 , E2 = 1 , E3 = 2
E4 = 6 , E5 = 3 , E6 = 11
E10 = 7 , E11 = 12
E12 = 13 , E13 = 14 , E14 = 15 , E15 = 8
E16 = 9 , E17 = 10 , E18 = 16
E20 = 0

; GROUPS:

G0 = 2560+ 0+ 2<4
G1 = 2560+ 1+ 2<4 , G2 = 2560+ 2+ 3<4
G3 = 2560+ 3+ 4<4 , G4 = 2560+ 4+ 5<4
G5 = 2560+ 5+ 6<4 , G6 = 2560+ 6+ 6<4
G7 = 2560+ 7+ 7<4 , G8 = 2560+ 8+ 8<4
G9 = 2560+ 9+ 9<4 , G10= 2560+10+10<4
G11= 2560+11+10<4 , G12= 2560+12+11<4
G13= 2560+13+12<4

; CR:

C0 = 2048+1024

; ENDS:

C1 = 2048+1 , C2 = 2048+2 , C3 = 2048+4
C4 = 2048+8 , C5 = 2048+16

; PUNCH OFF CODE FOR THE MACRO ASSEMBLER:

H C7:

; A/D SCAN TABLE PART 1

	C0	; INITIAL CR
	G1	; GROUP 1
A1:	J3 + E0	; PR4812A: L
	259, 614	; TERM, C1
A2:	J3 + E20	; TR 4811: L
	262, 102, -20	; TERM, C1, C2
A3:	J3 + J4 + E1	; FR 4811 : L, I1
	256, 1296, A1, A2	; TERM, C1, PR4812A, TR4811
A4:	J3 + J4 + E1	; FR4812 : L, I2
	257, 648, A1, A2	; TERM, C1, PR4812A, TR4811
A5:	J3 + J4 + E1	; FR4813 : L, I3
	258, 324, A1, A2	; TERM, C1, PR4812A, TR4811
A6:	J1 + J3 + E0	; PI4817 : L, H
	375	; HI
	261, 410	; TERM, C1
A7:	J3 + E12	; TI4801-6 : L
	65	; TERM
A8:	J3 + E0	; PRC 4816 : L
B1:	260, 1024	; TERM, C1
A9:	J3 + E12	; TI 4801-5 : L
	64	; TERM
A10:	J3 + E0	; PR 1107 : L
B2:	272, 410	; TERM, C1
A11:	J3 + E20	; TR 1101 : L
	273, 205, 50	; TERM, C1, C2
A12:	J3 + E0	; PRC 4804 : L
B3:	269, 2048	; TERM, C1
A13:	J3 + E12	; TI 4801-2 : L
	67	; TERM
A14:	J3 + J4 + E3	; FI 4801 : L, I4
	263, 248, 50, B3	; TERM, C1, C2, PRC 4804
A15:	J3 + J4 + E2	; FI 4804 : L, I5
	266, 642, 50, B3, A13	; TERM, C1, C2, PRC 4804, TI 4801-2
A16:	J1 + J2 + J3 + E0	; PRC 4811 : L, H, L
	340, 280	; HI, LO
B4:	271, 410	; TERM, C1
A17:	J3 + E12	; TI 4801-1 : L
	66	; TERM
A18:	J3 + J4 + E2	; FI 4802 : L, I6
	264, 350, 25, B4, A17	; TERM, C, C2, PRC 4811, TI 4801-1
A19:	J3 + J4 + E2	; FI 4803 : L, I7
	265, 667, 25, B4, A17	; TERM, C1, C2, PRC 4811, TI 4801-1
A20:	J3 + J4 + E2	; FI 4805: L, I8
	267, 3444, 25, B4, A17	; TERM, C1, C2, PRC 4811, TI 4801-1
A21:	J2 + J3 + E0	; PR 4808: L, L
	270	; LO
	270, 410	; TERM, C1
A22:	J3 + E0	; PRC 4801: L
B5:	268, 410	; TERM, C1

	C0 ~ G2	; CR, GROUP 2
A23:	J2 + J3 + J4 + E1	; 1 FRC 1301 : L, I9, L
	1000	; LO
	288, 3131, A6, A7	; TERM, C1, PI 4817, TI 4801-6
A24:	J2 + J3 + E2	; 1 FRC 1302 : L, L
	140	; LO
	289, 344, 25, B2, A11	; TERM, C1, C2, PR 1107, TR 1101
A25:	J3 + J4 + E3	; 1 FR 1304 : L, I12
	290, 263, 25, B4	; TERM, C1, C2, PRC 4811
A26:	J3 + E0	; 1 PRC 1301 : L
B6:	295, 512	; TERM, C1
A27:	J2 + J3 + J4 + E3	; 1 FR 1305 : L, I15, L
	600	; LO
	291, 1810, 20, B6	; TERM, C1, C2, 1 PRC 1301
A28:	J1 + J3 + E20	; 1 TRC 1304 : L, H
	480	; HI
	297, 307, 300	; TERM, C1, C2
A29:	J2 + J3 + J4 + E3	; 1 FRC 1306 : L, I18, L
	370	; LO
	292, 673, 20, B6	; TERM, C1, C2, 1 PRC 1301
A30:	J1 + J3 + J4 + E2	; 1 FRC 1308 : L, I87, H
	1000	; HI
	293, 1391, 100, B1, A9	; TERM, C1, C2, PRC 4816, TI 4801-5
A31:	J1 + J3 + E14	; 1 TR 1302-2 : L, H
	575	; HI
	33	; TERM
A32:	J3 + E0	; 1 PR 1306 : L
	296, 512	; TERM, C1
A33:	J3 + E13	; 1 TI 1301-8 : L
	163	; TERM
A34:	J3 + E13	; 1 TI 1301-9 : L
	164	; TERM
A35:	J1 + J3 + E13	; 1 TI 1301-10 : L, H
	800	; HI
	165	; TERM
A36:	J1 + J3 + E13	; 1 TI 1301-11 : L, H
	830	; HI
	166	; TERM
A37:	J1 + J3 + E13	; 1 TI 1301-19 : L, H
	830	; HI
	167	; TERM
A38:	J3 + E13	; 1 TI 1301-38 : L
	168	; TERM
A39:	J3 + E13	; 1 TI 1301-39 : L
	169	; TERM
A40:	J1 + J3 + E13	; 1 TI 1301-40 : L, H
	800	; HI
	170	; TERM
A41:	J1 + J3 + E13	; 1 TI 1301-41 : L, H

830	; HI
171	; TERM
A42: J1 + J3 + E13	; 1 TI 1301-49 : L, H
830	; HI
172	; TERM
A43: J1 + J3 + E14	; 1 TR 1302-3 : L, H
1100	; HI
34	; TERM
A44: J3 + E13	; 1 TI 1301-6 : L
162	; TERM
A45: J1 + J2 + J3 + E0	; 1 AR 1301 : L, H, L
250, 100	; HI, LO
0, 1024	; TERM, C1
A46: J1 + J3 + E14	; 1 TR 1302-1 : L, H
575	; HI
32	; TERM
A47: J1 + J3 + E14	; 1 TR 1302-4 : L, H
830	; HI
35	; TERM
A48: J3 + E0	; 1 AR 1302 : L
1, 3072	; TERM, C1
A49: J3 + E14	; 1 TR 1302-6 : L
36	; TERM
A50: J1 + J3 + E14	; 1 TR 1302-8 : L, H
1030	; HI
37	; TERM
A51: J1 + J3 + E0	; 1 AR 1303 : L, H
400	; HI
2, 1024	; TERM, C1
A52: J1 + J3 + E20	; 1 TRC 1306 : L, H
390	; HI
298, 205, 250	; TERM, C1, C2
C0 ~ G3	; CR, GROUP 3
A53: J2 + J3 + J4 + E1	; 2 FRC 1301 : L, I10, L
1000	; LO
320, 3131, A6, A7	; TERM, C1, PI 4817, TI 4801-6
A54: J2 + J3 + E2	; 2 FRC 1302 : L, L
140	; LO
321, 344, 25, B2, A11	; TERM, C1, C2, PR 1107, TR 1101
A55: J3 + J4 + E3	; 2 FR 1304 : L, I13
322, 263, 25, B4	; TERM, C1, C2, PRC 4811
A56: J3 + E0	; 2 PRC 1301 : L
B7: 327, 512	; TERM, C1
A57: J2 + J3 + J4 + E3	; 2 FR 1305 : L, I16, L
600	; LO
323, 1810, 20, B7	; TERM, C1, C2, 2 PRC 1301
A58: J1 + J3 + E20	; 2 TRC 1304 : L, H
480	; HI
329, 307, 300	; TERM, C1, C2

A59: J2 + J3 + J4 + E3 ; 2 FRC 1306 : L, I19, L
370 ; LO
324, 673, 20, B7 ; TERM, C1, C2, 2 PRC 1301
A60: J1 + J3 + J4 + E2 ; 2 FRC 1308 : L, I88, H
1000 ; HI
325, 1391, 100, B1, A9 ; TERM, C1, C2, PRC 4816, TI 4801-5
A61: J1 + J3 + E14 ; 2 TR 1302-2 : L, H
575 ; HI
39 ; TERM
A62: J3 + E0 ; 2 PR 1306 : L
328, 512 ; TERM, C1
A63: J3 + E13 ; 2 TI 1301-8 : L
175 ; TERM
A64: J3 + E13 ; 2 TI 1301-9 : L
176 ; TERM
A65: J1 + J3 + E13 ; 2 TI 1301-10 : L, H
800 ; HI
177 ; TERM
A66: J1 + J3 + E13 ; 2 TI 1301-11 : L, H
830 ; HI
178 ; TERM
A67: J1 + J3 + E13 ; 2 TI 1301-19 : L, H
830 ; HI
179 ; TERM
A68: J3 + E13 ; 2 TI 1301-38 : L
180 ; TERM
A69: J3 + E13 ; 2 TI 1301-39 : L
181 ; TERM
A70: J1 + J3 + E13 ; 2 TI 1301-40 : L, H
800 ; HI
182 ; TERM
A71: J1 + J3 + E13 ; 2 TI 1301-41 : L, H
830 ; HI
183 ; TERM
A72: J1 + J3 + E13 ; 2 TI 1301-49 : L, H
830 ; HI
184 ; TERM
A73: J1 + J3 + E14 ; 2 TR 1302-3 : L, H
1100 ; HI
40 ; TERM
A74: J3 + E13 ; 2 TI 1301-6 : L
174 ; TERM
A75: J1 + J2 + J3 + E0 ; 2 AR 1301 : L, H, L
250, 100 ; HI, LO
7, 1024 ; TERM, C1
A76: J1 + J3 + E14 ; 2 TR 1302-1 : L, H
575 ; HI
38 ; TERM
A77: J1 + J3 + E14 ; 2 TR 1302-4 : L, H

830	; HI
41	; TERM
A78: J3 + E0	; 2 AR 1302 : L
8, 3072	; TERM, C1
A79: J3 + E14	; 2 TR 1302-6 : L
42	; TERM
A80: J1 + J3 + E14	; 2 TR 1302-8 : L, H
1030	; HI
43	; TERM
A81: J1 + J3 + E0	; 2 AR 1303 : L, H
400	; HI
9, 1024	; TERM, C1
A82: J1 + J3 + E20	; 2 TRC 1306 : L, H
390	; HI
330, 205, 250	; TERM, C1, C2
C0 ~ G4	; CR, GROUP 4
A83: J2 + J3 + J4 + E1	; 3 FRC 1301: L, I11, L
1000	; LO
352, 3131, A6, A7	; TERM, C1, PI 4817, TI 4801-6
A84: J2 + J3 + E2	; 3 FRC 1302 : L, L
140	; LO
353, 344, 25, B2, A11	; TERM, C1, C2, PR 1107, TR 1101
A85: J3 + J4 + E3	; 3 FR 1304 : L, I14
354, 263, 25, B4	; TERM, C1, C2, PRC 4811
A86: J3 + E0	; 3 PRC 1301 : L
B8: 359, 512	; TERM, C1
A87: J2 + J3 + J4 + E3	; 3 FR 1305 : L, I17, L
600	; LO
355, 1810, 20, B8	; TERM, C1, C2, 3 PRC 1301
A88: J1 + J3 + E20	; 3 TRC 1304 : L, H
480	; HI
361, 307, 300	; TERM, C1, C2
A89: J2 + J3 + J4 + E3	; 3 FRC 1306 : L, I20, L
370	; LO
356, 673, 20, B8	; TERM, C1, C2, 3 PRC 1301
A90: J1 + J3 + J4 + E2	; 3 FRC 1308 : L, I89, H
1000	; HI
357, 1391, 100, B1, A9	; TERM, C1, C2, PRC 4816, TI 4801-5
A91: J1 + J3 + E14	; 3 TR 1302-2 : L, H
575	; HI
45	; TERM
A92: J3 + E0	; 3 PR 1306 : L
360, 512	; TERM, C1
A93: J3 + E13	; 3 TI 1301-8 : L
187	; TERM
A94: J3 + E13	; 3 TI 1301-9 : L
188	; TERM
A95: J1 + J3 + E13	; 3 TI 1301-10 : L, H
800	; HI

189	; TERM
A96: J1 + J3 + E13	; 3 TI 1301-11 : L, H
830	; HI
190	; TERM
A97: J1 + J3 + E13	; 3 TI 1301-19 : L, H
830	; HI
191	; TERM
A98: J3 + E13	; 3 TI 1301-38 : L
192	; TERM
A99: J3 + E13	; 3 TI 1301-39 : L
193	; TERM
A100: J1 + J3 + E13	; 3 TI 1301-40 : L, H
800	; HI
194	; TERM
A101: J1 + J3 + E13	; 3 TI 1301 - 41 : L, H
830	; HI
195	; TERM
A102: J1 + J3 + E13	; 3 TI 1301 - 49 : L, H
830	; HI
196	; TERM
A103: J1 + J3 + E14	; 3 TR 1302 - 3 : L, H
1100	; HI
46	; TERM
A104: J3 + E13	; 3 TI 1301 - 6 : L
186	; TERM
A105: J1 + J2 + J3 + E0	; 3 AR 1301 : L, H, L
250, 100	; HI, LO
14, 1024	; TERM, C1
A106: J1 + J3 + E14	; 3 TR 1302 - 1 : L, H
575	; HI
44	; TERM
A107: J1 + J3 + E14	; 3 TR 1302 - 4 : L, H
830	; HI
47	; TERM
A108: J3 + E0	; 3 AR 1302 : L
15, 3072	; TERM, C1
A109: J3 + E14	; 3 TR 1302 - 6 : L
48	; TERM
A110: J1 + J3 + E14	; 3 TR 1302 - 8 : L, H
1030	; HI
49	; TERM
A111: J1 + J3 + E0	; 3 AR 1303 : L, H
400	; HI
16, 1024	; TERM, C1
A112: J1 + J3 + E20	; 3 TRC 1306 : L, H
390	; HI
362, 205, 250	; TERM, C1, C2
C0 v G2	; CR, GROUP 2

A113: J3 + E4 ; 1 FRC 1311 : L
294, 233 ; TERM, C1
A114: J3 + E13 ; 1 TI 1301 - 59 : L
173 ; TERM
A115: J1 + J3 + E12 ; 1 TI 1401 - 6 : L, H
440 ; HI
72 ; TERM

A116: J3 + E20 ; 1 TRC 1403 : L
302, 205, 100 ; TERM, C1, C2
A117: J1 + J3 + E12 ; 1 TI 1401 - 14 : L, H
280 ; HI
73 ; TERM
A118: J1 + J3 + E0 ; 1 AR 1401 : L, H
350 ; HI
3, 2048 ; TERM, C1
A119: J3 + E12 ; 1 TI 1401 - 17 : L
74 ; TERM
A120: J3 + E12 ; 1 TI 1401 - 19 : L
75 ; TERM
A121: J3 + E0 ; 1 PIC 1407 : L
B9: 301, 512 ; TERM, C1
A122: J3 + J4 + E3 ; 1 FI 1403 : L, I21
299, 1254, 200, B9 ; TERM, C1, C2, 1 PIC 1407
A123: J3 + J4 + E3 ; 1 FI 1506 : L, I24
300, 1122, 250, B5 ; TERM, C1, C2, PRC 4801
A124: J3 + E12 ; 1 TR 1502 - 5 : L
76 ; TERM
A125: J3 + E4 ; 1 FIC 1502 : L
303, 1822 ; TERM, C1
A126: J3 + E4 ; 1 FIC 1503 : L
304, 557 ; TERM, C1
A127: J3 + E12 ; 1 TR 1502 - 1 : L
77 ; TERM
A128: J1 + J3 + E0 ; 1 AR 1501 : L, H
200 ; HI
4, 1024 ; TERM, C1
A129: J3 + E20 ; 1 TRC 1505 : L
307, 205, 250 ; TERM, C1, C2
A130: J1 + J3 + E12 ; 1 TR 1502 - 3 : L, H
375 ; HI
78 ; TERM
A131: J1 + J3 + E0 ; 1 AR 1502 : L, H
50 ; HI
5, 1024 ; TERM, C1

A132: J3 + E20 ; 1 AR 1504 : L
6, 410, 600 ; TERM, C1, C2
A133: J1 + J2 + J3 + E0 ; 1 PI 1518 : L, H, L

	275, 235	; HI, LO
B10:	306, 410	; TERM, C1
A134:	J1 + J3 + E12	; 1 TR 1502 - 4 : L, H
	50	; HI
	79	; TERM
A135:	J3 + E2	; 1 FI 1504 : L
	305, 961, 25, B10, A134	; TERM, C1, C2, 1PI 1518, 1 TR 1502 - 4
A136:	J3 + E12	; 1 TI 1602 - 24 : L
	100	; TERM
A137:	J3 + E0	; 1 PR 1603 : L
B11:	314, 410	; TERM, C1
A138:	J2 + J3 + E2	; 1 FI 1602 : L, L
	210	; LO
	309, 378, 3, B11, A139	; TERM, C1, C2, 1 PR 1603, 1 TI 1602 - 25
A139:	J3 + E12	; 1 TI 1602 - 25 : L
	101	; TERM
A140:	J3 + E12	; 1 TR 1601 - 1 : L
	80	; TERM
A141:	J3 + E12	; 1 TR 1601 - 2 : L
	81	; TERM
A142:	J3 + E12	; 1 TR 1601 - 3 : L
	82	; TERM
	C0 ~ G3	; CR, GROUP 3
A143:	J3 + E4	; 2 FRC 1311 : L
	326, 233	; TERM, C1
A144:	J3 + E13	; 2 TI 1301 - 59 : L
	185	; TERM
A145:	J1 + J3 + E12	; 2 TI 1401 - 6 : L, H
	440	; HI
	102	; TERM
A146:	J3 + E20	; 2 TRC 1403 : L
	334, 205, 100	; TERM, C1, C2
A147:	J1 + J3 + E12	; 2 TI 1401 - 14 : L, H
	280	; HI
	103	; TERM
A148:	J1 + J3 + E0	; 2 AR 1401 : L, H
	350	; HI
	10, 2048	; TERM, C1
A149:	J3 + E12	; 2 TI 1401 - 17 : L
	104	; TERM
A150:	J3 + E12	; 2 TI 1401 - 19 : L
	105	; TERM
A151:	J3 + E0	; 2 PIC 1407 : L
B12:	333, 512	; TERM, C1
A152:	J3 + J4 + E3	; 2 FI 1403 : L, I22
	331, 1254, 200, B12	; TERM, C1, C2, 2 PIC 1407
A153:	J3 + J4 + E3	; 2 FI 1506 : L, I25
	332, 1122, 250, B5	; TERM, C1, C2, PRC 4801

A154:	J3 + E12	; 2 TR 1502 - 5 : L
	106	; TERM
A155:	J3 + E4	; 2 FIC 1502 : L
	335, 1822	; TERM, C1
A156:	J3 + E4	; 2 FIC 1503 : L
	336, 557	; TERM, C1
A157:	J3 + E12	; 2 TR 1502 - 1 : L
	107	; TERM
A158:	J1 + J3 + E0	; 2 AR 1501 : L, H
	200	; HI
	11, 1024	; TERM, C1
A159:	J3 + E20	; 2 TRC 1505 : L
	339, 205, 250	; TERM, C1, C2
A160:	J1 + J3 + E12	; 2 TR 1502 - 3 : L, H
	375	; HI
	108	; TERM
A161:	J1 + J3 + E0	; 2 AR 1502 : L, H
	50	; HI
	12, 1024	; TERM, C1
A162:	J3 + E20	; 2 AR 1504 : L
	13, 410, 600	; TERM, C1, C2
A163:	J1 + J2 + J3 + E0	; 2 PI 1518 : L, H, L
	275, 235	; HI, LO
B13:	338, 410	; TERM, C1
A164:	J1 + J3 + E12	; 2 TR 1502 - 4 : L, H
	50	; HI
	109	; TERM
A165:	J3 + E2	; 2 FI 1504 : L
	337, 961, 25, B13, A164	; TERM, C1, C2, 2 PI 1518, 2 TR 1502 - 4
A166:	J3 + E12	; 2 TI 1602 - 24 : L
	130	; TERM
A167:	J3 + E0	; 2 PR 1603 : L
B14:	346, 410	; TERM, C1
A168:	J2 + J3 + E2	; 2 FI 1602 : L, L
	210	; LO
	341, 378, 3, B14, A169	; TERM, C1, C2, 2 PR 1603, 2 TI 1602 - 25
A169:	J3 + E12	; 2 TI 1602 - 25 : L
	131	; TERM
A170:	J3 + E12	; 2 TR 1601 - 1 : L
	110	; TERM
A171:	J3 + E12	; 2 TR 1601 - 2 : L
	111	; TERM
A172:	J3 + E12	; 2 TR 1601 - 3 : L
	112	; TERM
	C0 ~ G4	; CR, GROUP 4
A173:	J3 + E4	; 3 FRC 1311 : L
	358, 233	; TERM, C1
A174:	J3 + E13	; 3 TI 1301 - 59 : L

A175:	197 J1 + J3 + E12 440 132	; TERM ; 3 TI 1401 - 6 : L, H ; HI ; TERM
A176:	J3 + E20 366, 205, 100	; 3 TRC 1403 : L ; TERM, C1, C2
A177:	J1 + J3 + E12 280 133	; 3 TI 1401 - 14 : L, H ; HI ; TERM
A178:	J1 + J3 + E0 350 17, 2048	; 3 AR 1401 : L, H ; HI ; TERM, C1
A179:	J3 + E12 134	; 3 TI 1401 - 17 : L ; TERM
A180:	J3 + E12 135	; 3 TI 1401 - 19 : L ; TERM
A181:	J3 + E0	; 3 PIC 1407 : L
B15:	365, 512	; TERM, C1
A182:	J3 + J4 + E3 363, 1254, 200, B15	; 3 FI 1403 : L, I23 ; TERM, C1, C2, 3 PIC 1407
A183:	J3 + J4 + E3 364, 1122, 250, B5	; 3 FI 1506 : L, I26 ; TERM, C1, C2, PRC 4801
A184:	J3 + E12 136	; 3 TR 1502 - 5 : L ; TERM
A185:	J3 + E4 367, 1822	; 3 FIC 1502 : L ; TERM, C1
A186:	J3 + E4 368, 557	; 3 FIC 1503 : L ; TERM, C1
A187:	J3 + E12 137	; 3 TR 1502 - 1 : L ; TERM
A188:	J1 + J3 + E0 200 18, 1024	; 3 AR 1501 : L, H ; HI ; TERM, C1
A189:	J3 + E20 371, 205, 250	; 3 TRC 1505 : L ; TERM, C1, C2
A190:	J1 + J3 + E12 375 138	; 3 TR 1502 - 3 : L, H ; HI ; TERM
A191:	J1 + J3 + E0 50 19, 1024	; 3 AR 1502 : L, H ; HI ; TERM, C1
A192:	J3 + E20 20, 410, 600	; 3 AR 1504 : L ; TERM, C1, C2
A193:	J1 + J2 + J3 + E0 275, 235	; 3 PI 1518 : L, H, L ; HI, LO
B16:	370, 410	; TERM, C1
A194:	J1 + J3 + E12 50	; 3 TR 1502 - 4 : L, H ; HI

```

139 ; TERM
A195: J3 + E2 ; 3 FI 1504 : L
      369, 961, 25, B16, A194 ; TERM, C1, C2, 2 PI 1518, 3 TR 1502 - 4
A196: J3 + E12 ; 3 TI 1602 - 24 : L
      160 ; TERM
A197: J3 + E0 ; 3 PR 1603 : L
B17: 378, 410 ; TERM, C1
A198: J2 + J3 + E2 ; 3 FI 1602 : L, L
      210 ; LO
      373, 378, 3, B17, A199 ; TERM, C1, C2, 3 PR 1603, 3 TI 1602 - 25
A199: J3 + E12 ; 3 TI 1602 - 25 : L
      161 ; TERM
A200: J3 + E12 ; 3 TR 1601 - 1 : L
      140 ; TERM
A201: J3 + E12 ; 3 TR 1601 - 2 : L
      141 ; TERM
A202: J3 + E12 ; 3 TR 1601 - 3 : L
      142 ; TERM
      C0 ~ G2 ; CR, GROUP 2
A203: J3 + E12 ; 1 TR 1601 - 4 : L
      83 ; TERM
A204: J3 + E12 ; 1 TR 1601 - 5 : L
      84 ; TERM
A205: J1 + J2 + J3 + E12 ; 1 TI 1602 - 1 : L, H, L
      450, 370 ; HI, LO
      86 ; TERM
A206: J1 + J3 + E12 ; 1 TI 1602 - 2 : L, H
      550 ; HI
      87 ; TERM
A207: J1 + J3 + E12 ; 1 TI 1602 - 3 : L, H
      550 ; HI
      88 ; TERM
A208: J1 + J3 + E12 ; 1 TI 1602 - 4 : L, H
      550 ; HI
      89 ; TERM
A209: J1 + J3 + E12 ; 1 TI 1602 - 5 : L, H
      550 ; HI
      90 ; TERM
A210: J1 + J3 + E12 ; 1 TI 1602 - 6 : L, H
      550 ; HI
      91 ; TERM
A211: J1 + J3 + E12 ; 1 TI 1602 - 7 : L, H
      550 ; HI
      92 ; TERM
A212: J1 + J3 + E12 ; 1 TI 1602 - 8 : L, H
      550 ; HI
      93 ; TERM

```

A213:	J1 + J3 + E12	; 1 TI 1602 - 9 : L, H
	550	; HI
	94	; TERM
A214:	J1 + J3 + E12	; 1 TI 1602 - 10 : L, H
	550	; HI
	95	; TERM
A215:	J1 + J3 + E12	; 1 TI 1602 - 11 : L, H
	550	; HI
	96	; TERM
A216:	J1 + J3 + E12	; 1 TI 1602 - 12 : L, H
	550	; HI
	97	; TERM
A217:	J1 + J3 + E12	; 1 TI 1602 - 13 : L, H
	550	; HI
	98	; TERM
A218:	J1 + J3 + E12	; 1 TR 1601 - 6 : L, H
	370	; HI
	85	; TERM
A219:	J2 + J3 + J4 + E3	; 1 FI 1601 : L, I27, L
	170	; LO
	308, 316, 50, B3	; TERM, C1, C2, PRC 4804
A220:	J3 + E12	; 1 TI 1602 - 29 : L
	99	; TERM
A221:	J3 + J4 + E4	; 1 FRC 1603 : L, I30
	310, 506	; TERM, C1
A222:	J3 + J4 + E4	; 1 FI 1604 : L, I33
	311, 1012	; TERM, C1
A223:	J3 + J4 + E4	; 1 FR 1605 : L, I36
	312, 253	; TERM, C1
A224:	J3 + E4	; 1 FI 1606 : L
	313, 1619	; TERM, C1
	C0 ~ G3	; CR, GROUP 3
A225:	J3 + E12	; 2 TR 1601 - 4 : L
	113	; TERM
A226:	J3 + E12	; 2 TR 1601 - 5 : L
	114	; TERM
A227:	J1 + J2 + J3 + E12	; 2 TI 1602 - 1 : L, H, L
	450, 370	; HI, LO
	116	; TERM
A228:	J1 + J3 + E12	; 2 TI 1602 - 2 : L, H
	550	; HI
	117	; TERM
A229:	J1 + J3 + E12	; 2 TI 1602 - 3 : L, H
	550	; HI
	118	; TERM
A230:	J1 + J3 + E12	; 2 TI 1602 - 4 : L, H
	550	; HI
	119	; TERM

A231: J1 + J3 + E12 ; 2 TI 1602 - 5 : L, H
550 ; HI
120 ; TERM
A232: J1 + J3 + E12 ; 2 TI 1602 - 6 : L, H
550 ; HI
121 ; TERM
A233: J1 + J3 + E12 ; 2 TI 1602 - 7 : L, H
550 ; HI
122 ; TERM
A234: J1 + J3 + E12 ; 2 TI 1602 - 8 : L, H
550 ; HI
123 ; TERM
A235: J1 + J3 + E12 ; 2 TI 1602 - 9 : L, H
550 ; HI
124 ; TERM
A236: J1 + J3 + E12 ; 2 TI 1602 - 10 : L, H
550 ; HI
125 ; TERM
A237: J1 + J3 + E12 ; 2 TI 1602 - 11 : L, H
550 ; HI
126 ; TERM
A238: J1 + J3 + E12 ; 2 TI 1602 - 12 : L, H
550 ; HI
127 ; TERM
A239: J1 + J3 + E12 ; 2 TI 1602 - 13 : L, H
550 ; HI
128 ; TERM
A240: J1 + J3 + E12 ; 2 TR 1601 - 6 : L, H
370 ; HI
115 ; TERM
A241: J2 + J3 + J4 + E3 ; 2 FI 1601 : L, I28, L
170 ; LO
340, 316, 50, 83 ; TERM, C1, C2, PRC 4804
A242: J3 + E12 ; 2 TI 1602 - 29 : L
129 ; TERM
A243: J3 + J4 + E4 ; 2 FRC 1603 : L, I31
342, 506 ; TERM, C1
A244: J3 + J4 + E4 ; 2 FI 1604 : L, I34
343, 1012 ; TERM, C1
A245: J3 + J4 + E4 ; 2 FR 1605 : L, I37
344, 253 ; TERM, C1
A246: J3 + E4 ; 2 FI 1606 : L
345, 1619 ; TERM, C1
C0 ~ G4 ; CR, GROUP 4
A247: J3 + E12 ; 3 TR 1601 - 4 : L
143 ; TERM
A248: J3 + E12 ; 3 TR 1601 - 5 : L
144 ; TERM

A249: J1 + J2 + J3 + E12 ; 3 TI 1602 - 1 : L, H, L
450, 370 ; HI, LO
146 ; TERM
A250: J1 + J3 + E12 ; 3 TI 1602 - 2 : L, H
550 ; HI
147 ; TERM
A251: J1 + J3 + E12 ; 3 TI 1602 - 3 : L, H
550 ; HI
148 ; TERM
A252: J1 + J3 + E12 ; 3 TI 1602 - 4 : L, H
550 ; HI
149 ; TERM
A253: J1 + J3 + E12 ; 3 TI 1602 - 5 : L, H
550 ; HI
150 ; TERM
A254: J1 + J3 + E12 ; 3 TI 1602 - 6 : L, H
550 ; HI
151 ; TERM
A255: J1 + J3 + E12 ; 3 TI 1602 - 7 : L, H
550 ; HI
152 ; TERM
A256: J1 + J3 + E12 ; 3 TI 1602 - 8 : L, H
550 ; HI
153 ; TERM
A257: J1 + J3 + E12 ; 3 TI 1602 - 9 : L, H
550 ; HI
154 ; TERM
A258: J1 + J3 + E12 ; 3 TI 1602 - 10 : L, H
550 ; HI
155 ; TERM
A259: J1 + J3 + E12 ; 3 TI 1602 - 11 : L, H
550 ; HI
156 ; TERM
A260: J1 + J3 + E12 ; 3 TI 1602 - 12 : L, H
550 ; HI
157 ; TERM
A261: J1 + J3 + E12 ; 3 TI 1602 - 13 : L, H
550 ; HI
158 ; TERM
A262: J1 + J3 + E12 ; 3 TR 1601 - 6 : L, H
370 ; HI
145 ; TERM
A263: J2 + J3 + J4 + E3 ; 3 FI 1601 : L, I29, L
170 ; LO
372, 316, 50, B3 ; TERM, C1, C2, PRC 4804
A264: J3 + E12 ; 3 TI 1602 - 29 : L
159 ; TERM
A265: J3 + J4 + E4 ; 3 FRC 1603 : L, I32

A266:	374, 506 J3 + J4 + E4	; TERM, C1 ; 3 FI 1604 : L, 135
A267:	375, 1012 J3 + J4 + E4	; TERM, C1 ; 3 FR 1605 : L, 138
A268:	376, 253 J3 + E4	; TERM, C1 ; 3 FI 1606 : L
A269:	377, 1619 C0 ~ G1 J3 + E0	; TERM, C1 ; CR, GROUP 1 ; PRC 1701 : L
A270:	281, 256 J3 + E0	; TERM, C1 ; PRC 1705 : L
B18:	284, 1638	; TERM, C1
A271:	J3 + E12 71	; TI 1701 - 7 : L ; TERM
A272:	J3 + E2 278, 2442, 63, B18, A271	; FI 1705 : L ; TERM, C1, C2, PRC 1705, TI 1701 - 7
A273:	J3 + J4 + E3 277, 801, 50, B3	; FRC 1702, RANGE 1 : L, 139 ; TERM, C1, C2, PRC 4804
A274:	J3 + J4 + J5 + E3 277, 1832, 50, B3	; FRC 1702, RANGE 2 : L, 140, REMOVED ; TERM, C1, C2, PRC 4804
A275:	J3 + E0 283, 256	; PRC 1710 : L ; TERM, C1
A276:	J3 + J4 + E4 279, 304	; FI 1706 : L, 141 ; TERM, C1
A277:	J3 + J4 + E4 280, 1113	; FI 1709 : L, 142 ; TERM, C1
A278:	J3 + J5 + E4 284, 243	; FI 1801, RANGE 1 : L, REMOVED ; TERM, C1
A279:	J3 + E4 284, 607	; FI 1801, RANGE 2 : L ; TERM, C1
A280:	J3 + J5 + E4 284, 1518	; FI 1801, RANGE 3: L, REMOVED ; TERM, C1
A281:	J3 + E0 285, 1024	; TRC 1805 : L ; TERM, C1
A282:	J3 + E6 224	; LI 1901 A : L ; TERM
A283:	J3 + E6 225	; LI 1901 B : L ; TERM
A284:	J3 + E6 226	; LI 1901 C : L ; TERM
A285:	J3 + E12 68	; TI 4801 - 7 : L ; TERM
A286:	J3 + J4 + E4 275, 455	; FI 4821 : L, 143 ; TERM, C1
A287:	J4 + E15 275, 455, A290, A285	; KC 4821 : 144 ; TERM, C1, TI 4801 - 9, TI 4801 - 7

A288: J3 + J4 + E4 ; FI 4822 : L, I45
276, 658 ; TERM, C1
A289: J4 + E15 ; KC 4822 : I46
276, 658, A291, A285 ; TERM, C1, TI 4801 - 10, TI 4801 - 7
A290: J3 + E12 ; TI 4801 - 9 : L
69 ; TERM
A291: J3 + E12 ; TI 4801 - 10 : L
70 ; TERM
A292: J3 + J4 + E0 ; FI 4301 : L, I47
286, 375 ; TERM, C1
A293: J3 + J4 + E0 ; FI 4401 : L, I48
287, 375 ; TERM, C1
A294: J3 + J4 + E4 ; FI 4501 : L, I49
274, 708 ; TERM, C1
A295: E18 ; DUMMIES
A296: E18
A297: E18
A298: E18
A299: E18
A300: E18
C1 ;

; END PART 1

L
; END S11 FIRST PART
N
T


```

C8:      C0      ; AD SCAN TABLE PART 2
          G6      ; GROUP 6
A301:    J1 + J2 + J3 + E0 ; 1 PRC 2106 : L, H, L
          440, 300 ; HI, LO
B19:     545, 1024 ; TERM, C1
A302:    J3 + E12 ; 1 TD 2101 : L
          416      ; TERM
A303:    J3 + J4 + E2 ; 1 FRRC 2101 : L, I50
          544, 2179, 100, B19, A302 ; TERM, C1, C2, 1 PRC 2106, 1 TD 2101
A304:    J3 + E0 ; 1 FI 2101 : L
          554, 1331 ; TERM, C1
A305:    J3 + E12 ; 1 TD 2102 : L
          422      ; TERM
A306:    J1 + J2 + J3 + E14 ; 1 TD 2104 : L, H, L
          930, 550 ; HI, LO
          384      ; TERM
A307:    J1 + J2 + J3 + E14 ; 1 TD 2106 : L, H, L
          930, 550 ; HI, LO
          385      ; TERM
A308:    J1 + J2 + J3 + E14 ; 1 TD 2108 : L, H, L
          930, 550 ; HI, LO
          386      ; TERM
A309:    J3 + E12 ; 1 TD 2105 : L
          419      ; TERM
A310:    J3 + E12 ; 1 TD 2107 : L
          420      ; TERM
A311:    J3 + E12 ; 1 TD 2109 : L
          421      ; TERM
A312:    J3 + E0 ; 1 PRC 2152 : L
B20:     549, 410 ; TERM, C1
A313:    J3 + E12 ; 1 TD 2103 : L
          418      ; TERM
A314:    J3 + J4 + E2 ; 1 FR 2102 : L, I54
          551, 862, 25, B20, A313 ; TERM, C1, C2, 1 PRC 2152, 1 TD 2103

A315:    J3 + J4 + E4 ; 1 FR 2103 : L, I58
          550, 607 ; TERM, C1
A316:    J3 + J4 + E2 ; 1 FR 2105 : L, I62
          548, 362, 25, B20, A313 ; TERM, C1, C2, 1 PRC 2152, 1 TD 2103
A317:    J3 + E0 ; 1 PDI 2118 : L
          556, 512 ; TERM, C1
A318:    J3 + E0 ; 1 FRC 2109 : L
          555, 1536 ; TERM, C1
A319:    J3 + E0 ; 1 FRC 2111 : L
          553, 2048 ; TERM, C1
A320:    J3 + E20 ; 1 DIR 2115 : L
          547, 205, 1200 ; TERM, C1, C2
A321:    J3 + E12 ; 1 TD 2110 : L
          417      ; TERM
A322:    J3 + E17 ; 1 WTC 2112 : L
          547, 1, A320, A321 ; TERM, C1, 1 DIR 2115, 1 TD 2110
A323:    J3 + J4 + E10 ; 1 FRC 2112 : L, I66
          546, 629, A320, A321 ; TERM, C1, 1 DIR 2115, 1 TD 2110
          G5      ; GROUP 5
A324:    J3 + E0 ; FI 2108 : L
          552, 512 ; TERM, C1
A325:    J3 + J4 + E4 ; FR 2110 : L, I70
          557, 1316 ; TERM, C1
A326:    J2 + J3 + J4 + E4 ; FRA 2117 : L, I71, L
          400      ; LO

```

	558, 708	; TERM, C1
A327:	J3 + E0	; PD 2102 : L
B21:	559, 1024	; TERM
A328:	J3 + E3	; FD 2111 : L
	560, 1471, 100, B21	; TERM, C1, C2, PD 2102
	C0 ~ G7	; CR, GROUP 7
A329:	J1 + J2 + J3 + E0	; 2 PRC 2106 : L, H, L
	440, 300	; HI, LO
B22:	562, 1024	; TERM, C1
A330:	J3 + E12	; 2 TD 2101 : L
	423	; TERM
A331:	J3 + J4 + E2	; 2 FRRC 2101 : L, I51
	561, 2179, 100, B22, A330	; TERM, C1, C2, 2 PRC 2106, 2 TD 2101
A332:	J3 + E0	; 2 FI 2101 : L
	570, 1331	; TERM, C1
A333:	J3 + E12	; 2 TD 2102 : L
	429	; TERM
A334:	J1 + J2 + J3+E14	; 2 TD 2104 : L, H, L
	930, 550	; HI, LO
	387	; TERM
A335:	J1 + J2 + J3 + E14	; 2 TD 2106 : L, H, L
	930, 550	; HI, LO
	388	; TERM
A336:	J1 + J2 + J3 + E14	; 2 TD 2108 : L, H, L
	930, 550	; HI, LO
	389	; TERM
A337:	J3 + E12	; 2 TD 2105 : L
	426	; TERM
A338:	J3 + E12	; 2 TD 2107 : L
	427	; TERM
A339:	J3 + E12	; 2 TD 2109 : L
	428	; TERM
A340:	J3 + E0	; 2 PRC 2152 : L
B23:	566, 410	; TERM, C1
A341:	J3 + E12	; 2 TD 2103 : L
	425	; TERM
A342:	J3 + J4 + E2	; 2 FR 2102 : L, I55
	568, 862, 25, B23, A341	; TERM, C1, C2, 2 PRC 2152, 2 TD 2103
A343:	J3 + J4 + E4	; 2 FR 2103 : L, I59
	567, 607	; TERM, C1
A344:	J3 + J4 + E2	; 2 FR 2105 : L, I63
	565, 362, 25, B23, A341	; TERM, C1, C2, 2 PRC 2152, 2 TD 2103
A345:	J3 + E0	; 2 PDI 2118 : L
	572, 512	; TERM, C1
A346:	J3 + E0	; 2 FRC 2109 : L
	571, 1536	; TERM, C1
A347:	J3 + E0	; 2 FRC 2111 : L
	569, 2048	; TERM, C1
A348:	J3 + E20	; 2 DIR 2115 : L
	564, 205, 1200	; TERM, C1, C2
A349:	J3 + E12	; 2 TD 2110 : L
	424	; TERM
A350:	J3 + E17	; 2 WTC 2112 : L
	564, 1, A348, A349	; TERM, C1, 2 DIR 2115, 2 TD 2110
A351:	J3 + J4 + E10	; 2 FRC 2112 : L, I67
	563, 629, A348, A349	; TERM, C1, 2 DIR 2115, 2 TD 2110
	C0 ~ G8	; CR, GROUP 8
A352:	J1 + J2 + J3 + E0	; 3 PRC 2106 : L, H, L

	440, 300	; HI, LO
B24:	577, 1024	; TERM, C1
A353:	J3 + E12	; 3 TD 2101 : L
	430	; TERM
A354:	J3 + J4 + E2	; 3 FRRC 2101 : L, I52
	576, 2179, 100, B24, A353	; TERM, C1, C2, 3 PRC 2106, 3 TD 2101
A355:	J3 + E0	; 3 FI 2101 : L
	585, 1331	; TERM, C1
A356:	J3 + E12	; 3 TD 2102 : L
	436	; TERM
A357:	J1 + J2 + J3 + E14	; 3 TD 2104 : L, H, L
	930, 550	; HI, LO
	390	; TERM
A358:	J1 + J2 + J3 + E14	; 3 TD 2106 : L, H, L
	930, 550	; HI, LO
	391	; TERM
A359:	J1 + J2 + J3 + E14	; 3 TD 2108 : L, H, L
	930, 550	; HI, LO
	392	; TERM
A360:	J3 + E12	; 3 TD 2105 : L
	433	; TERM
A361:	J3 + E12	; 3 TD 2107 : L
	434	; TERM
A362:	J3 + E12	; 3 TD 2109 : L
	435	; TERM
A363:	J3 + E0	; 3 PRC 2152 : L
B25:	581, 410	; TERM, C1
A364:	J3 + E12	; 3 TD 2103 : L
	432	; TERM
A365:	J3 + J4 + E2	; 3 FR 2102 : L, I56
	583, 862, 25, B25, A364	; TERM, C1, C2, 3 PRC 2152, 3 TD 2103
A366:	J3 + J4 + E4	; 3 FR 2103 : L, I60
	582, 607	; TERM, C1
A367:	J3 + J4 + E2	; 3 FR 2105 : L, I64
	580, 362, 25, B25, A364	; TERM, C1, C2, 3 PRC 2152, 3 TD 2103
A368:	J3 + E0	; 3 PDI 2118 : L
	587, 512	; TERM, C1
A369:	J3 + E0	; 3 FRC 2109 : L
	586, 1536	; TERM, C1
A370:	J3 + E0	; 3 FRC 2111 : L
	584, 2048	; TERM, C1
A371:	J3 + E20	; 3 DIR 2115 : L
	579, 205, 1200	; TERM, C1, C2
A372:	J3 + E12	; 3 TD 2110 : L
	431	; TERM
A373:	J3 + E17	; 3 WCT 2112 : L
	579, 1, A371, A372	; TERM, C1, 3 DIR 2115, 3 TD 2110
A374:	J3 + J4 + E10	; 3 FRC 2112 : L, I68
	578, 629, A371, A372	; TERM, C1, 3 DIR 2115, 3 TD 2110
	C0 ~ G9	; CR, GROUP 9
A375:	J1 + J2 + J3 + E0	; 4 PRC 2106 : L, H, L
	440, 300	; HI, LO
B26:	589, 1024	; TERM, C1
A376:	J3 + E12	; 4 TD 2101 : L
	437	; TERM
A377:	J3 + J4 + E2	; 4 FRRC 2101 : L, I53
	588, 2179, 100, B26, A376	; TERM, C1, C2, 3 PRC 2106, 3 TD 2101
A378:	J3 + E0	; 4 FI 2101 : L

597, 1331	; TERM, C1
A379: J3 + E12	; 4 TD 2102 : L
443	; TERM
A380: J1 + J2 + J3+E14	; 4 TD 2104 : L, H, L
930, 550	; HI, LO
393	; TERM
A381: J1 + J2 + J3 + E14	; 4 TD 2106 : L, H, L
930, 550	; HI, LO
394	; TERM
A382: J1 + J2 + J3 + E14	; 4TD 2108 : L, H, L
930, 550	; HI, LO
395	; TERM
A383: J3 + E12	; 4 TD 2105 : L
440	; TERM
A384: J3 + E12	; 4 TD 2107 : L
441	; TERM
A385: J3 + E12	; 4 TD 2109 : L
442	; TERM
A386: J3 + E0	; 4 PRC 2152 : L
B27: 593, 410	; TERM, C1
A387: J3 + E12	; 4 TD 2103 : L
439	; TERM
A388: J3 + J4 + E2	; 4 FR 2102 : L, I57
595, 862, 25, B27, A387	; TERM, C1, C2, 4 PRC 2152, 4 TD 2103
A389: J3 + J4 + E4	; 4 FR 2103 : L, I61
594, 607	; TERM, C1
A390: J3 + J4 + E2	; 4 FR 2105 : L, I65
592, 362, 25, B27, A387	; TERM, C1, C2, 4 PRC 2152, 4 TD 2103
A391: J3 + E0	; 4 PDI 2118 : L
599, 512	; TERM, C1
A392: J3 + E0	; 4 FRC 2109 : L
598, 1536	; TERM, C1
A393: J3 + E0	; 4 FRC 2111 : L
596, 2048	; TERM, C1
A394: J3 + E20	; 4 DIR 2115 : L
591, 205, 1200	; TERM, C1, C2
A395: J3 + E12	; 4 TD 2110 : L
438	; TERM
A396: J3 + E17	; 4 WTC 2112 : L
591, 1, A394, A395	; TERM, C1, 4 DIR 2115, 4 TD 2110
A397: J3 + J4 + E10	; 4 FRC 2112 : L, I69
590, 629, A394, A395	; TERM, C1, 4 DIR 2115, 4 TD 2110
C0 v G11	; CR, GROUP 11
A398: J1 + J2 + J3 + E0	; 1 PRC 3307 : L, H, L
750, 450	; HI, LO
B28: 617, 1024	; TERM, C1
A399: J3 + E0	; 1 TRC 3301 : L
625, 1536	; TERM, C1
A400: J3 + J4 + E5	; 1 FRTC 3101 : L, I72
608, 1887, 100, B28	; TERM, C1, C2, 1 PRC 3307
A401: J3 + J4 + E5	; 1 FRTC 3102 : L, I75
609, 1887, 100, B28	; TERM, C1, C2, 1 PRC 3307
A402: J1 + J2 + J3 + E0	; 1 PRC 3304 : L, H, L
420, 250	; HI, LO
618, 614	; TERM, C1
A403: J3 + J4 + E16	; 1 FRC 3103 : L, I78
496, 440, A422	; TERM, C1, DR 3001
A404: J3 + J4 + E16	; 1 FRC 3104 : L, I81
497, 440, A422	; TERM, C1, DR 3001

A405:	J3 + E20	; 1 TRC 3307 : L
	626, 819, -400	; TERM, C1, C2
A406:	J1 + J3+E0	; 1 TRC 3110 : L, H
	210	; HI
	627, 256	; TERM, C1
A407:	J1 + J3 + E0	; 1 TRC 3111 : L, H
	210	; HI
	628, 256	; TERM, C1
A408:	J2 + J3 + E0	; 1 PRC 3108 : L, L
	300	; LO
	619, 614	; TERM, C1
A409:	J2 + J3 + E0	; 1 PRC 3109 : L, L
	300	; LO
	620, 614	; TERM, C1
A410:	J3 + E0	; 1 FRC 3105 : L
	610, 230	; TERM, C1
A411:	J3 + E0	; 1 FRC 3106 : L
	611, 230	; TERM, C1
A412:	J3 + E20	; 1 PRC 3113 : L
	621, 1024, -500	; TERM, C1, C2
A413:	J3 + E20	; 1 PRC 3114 : L
	622, 1024, -500	; TERM, C1, C2
A414:	J3 + E0	; 1 PRC 3115 : L
	623, 1024	; TERM, C1
A415:	J3 + E0	; 1 PRC 3116 : L
	624, 1024	; TERM, C1
A416:	J3 + E0	; 1 TR 3124 : L
	629, 2048	; TERM, C1
A417:	J3 + J4 + E5	; 1 FRT 3228 : L, I84
	616, 2455, 63, B29	; TERM, C1, C2, PRC 3044
A418:	J3 + E11	; 1 TR 3234 : L
	464	; TERM
A419:	J3 + E11	; 1 TR 3235 : L
	465	; TERM
A420:	J3 + E11	; 1 TR 3236 : L
	466	; TERM
A421:	J3 + E11	; 1 TR 3237 : L
	467	; TERM
	G 10	; GROUP 10
A422:	J3 + E20	; DR 3001 : L
	636, 56, 1296	; TERM, C1, C2
A423:	E18	; DUMMY
A424:	J2 + J3 + E0	; PRC 3044 : L, L
	1000	; LO
B29:	634, 1619	; TERM, C1
A425:	J3 + E0	; TRC 3070 : L
	637, 256	; TERM, C1
A426:	J2 + J3 + E0	; PRC 3046 : L, L
	300	; LO
	635, 1619	; TERM, C1
	C0 v G12	; CR, GROUP 12
A427:	J1 + J2 + J3 + E0	; 2 PRC 3307 : L, H, L
	750, 450	; HI, LO
B30:	649, 1024	; TERM, C1
A428:	J3 + E0	; 2 TRC 3301 : L
	657, 1536	; TERM, C1
A429:	J3 + J4 + E5	; 2 FRTC 3101 : L, I73
	640, 1887, 100, B30	; TERM, C1, C2, 2 PRC 330

A430:	J3 + J4 + E5	; 2 FRTC 3102 : L, 176
	641, 1887, 100, B30	; TERM, C1, C2, 2 PRC 3307
A431:	J1 + J2 + J3 + E0	; 2 PRC 3304 : L, H, L
	420, 250	; HI, LO
	650, 614	; TERM, C1
A432:	J3 + J4 + E16	; 2 FRC 3103 : L, 179
	502, 440, A422	; TERM, C1, DR 3001
A433:	J3 + J4 + E16	; 2 FRC 3104 : L, 182
	503, 440, A422	; TERM, C1, DR 3001
A434:	J3 + E20	; 2 TRC 3307 : L
	658, 819, -400	; TERM, C1, C2
A435:	J1 + J3 + E0	; 2 TRC 3110 : L, H
	210	; HI
	659, 256	; TERM, C1
A436:	J1 + J3 + E0	; 2 TRC 3111 : L, H
	210	; HI
	660, 256	; TERM, C1
A437:	J2 + J3 + E0	; 2 PRC 3108 : L, L
	300	; LO
	651, 614	; TERM, C1
A438:	J2 + J3 + E0	; 2 PRC 3109 : L, L
	300	; LO
	652, 614	; TERM, C1
A439:	J3 + E0	; 2 FRC 3105 : L
	642, 230	; TERM, C1
A440:	J3 + E0	; 2 FRC 3106 : L
	643, 230	; TERM, C1
A441:	J3 + E20	; 2 PRC 3113 : L
	653, 1024, -500	; TERM, C1, C2
A442:	J3 + E20	; 2 PRC 3114 : L
	654, 1024, -500	; TERM, C1, C2
A443:	J3 + E0	; 2 PRC 3115 : L
	655, 1024	; TERM, C1
A444:	J3 + E0	; 2 PRC 3116 : L
	656, 1024	; TERM, C1
A445:	J3 + E0	; 2 TR 3124 : L
	661, 2048	; TERM, C1
A446:	J3 + J4 + E5	; 2 FRT 3228 : L, 185
	648, 2455, 63, B29	; TERM, C1, C2, PRC 3044
A447:	J3 + E11	; 2 TR 3234 : L
	457	; TERM
A448:	J3 + E11	; 2 TR 3235 : L
	458	; TERM
A449:	J3 + E11	; 2 TR 3236 : L
	459	; TERM
A450:	J3 + E11	; 2 TR 3237 : L
	460	; TERM
	C0 ~ G 13	; CR, GROUP 13
A451:	J1 + J2 + J3 + E0	; 3 PRC 3307 : L, H, L
	750, 450	; HI, LO
B31:	681, 1024	; TERM, C1
A452:	J3 + E0	; 3 TRC 3301 : L
	689, 1536	; TERM, C1
A453:	J3 + J4 + E5	; 3 FRTC 3101 : L, 174
	672, 1887, 100, B31	; TERM, C1, C2, 3 PRC 3307
A454:	J3 + J4 + E5	; 3 FRTC 3102 : L, 177
	673, 1887, 100, B31	; TERM, C1, C2, 3 PRC 3307
A455:	J1 + J2 + J3 + E0	; 3 PRC 3304 : L, H, L
	420, 250	; HI, LO

	682, 614		; TERM, C1
A456:	J3 + J4 + E16 508, 440, A422		; 3 FRC 3103 : L, I80 ; TERM, C1, DR 3001
A457:	J3 + J4 + E16 509, 440, A422		; 3 FRC 3104 : L, I83 ; TERM, C1, DR 3001
A458:	J3 + E20 690, 819, -400		; 3 TRC 3307 : L ; TERM, C1, C2
A459:	J1 + J3 + E0 210 691, 256		; 3 TRC 3110 : L, H ; HI ; TERM, C1
A460:	J1 + J3 + E0 210 692, 256		; 3 TRC 3111 : L, H ; HI ; TERM, C1
A461:	J2 + J3 + E0 300 683, 614		; 3 PRC 3108 : L, L ; LO ; TERM, C1
A462:	J2 + J3 + E0 300 684, 614		; 3 PRC 3109 : L, L ; LO ; TERM, C1
A463:	J3 + E0 674, 230		; 3 FRC 3105 : L ; TERM, C1
A464:	J3 + E0 675, 230		; 3 FRC 3106 : L ; TERM, C1
A465:	J3 + E20 685, 1024, -500		; 3 PRC 3113 : L ; TERM, C1, C2
A466:	J3 + E20 686, 1024, -500		; 3 PRC 3114 : L ; TERM, C1, C2
A467:	J3 + E0 687, 1024		; 3 PRC 3115 : L ; TERM, C1
A468:	J3 + E0 688, 1024		; 3 PRC 3116 : L ; TERM, C1
A469:	J3 + E0 693, 2048		; 3 TR 3124 : L ; TERM, C1
A470:	J3 + J4 + E5 680, 2455, 63, B29		; 3 FRT 3228 : L, I86 ; TERM, C1, C2, PRC 3044
A471:	J3 + E11 461		; 3 TR 3234 : L ; TERM
A472:	J3 + E11 462		; 3 TR 3235 : L ; TERM
A473:	J3 + E11 486		; 3 TR 3236 : L ; TERM
A474:	J3 + E11 487 C0 ~ G 11		; 3 TR 3237 : L ; TERM ; CR, GROUP 11
A475:	J3 + E0 498, 1536		; 1 FRTC 3212 : L ; TERM, C1
A476:	J3 + E0 499, 1536		; 1 FRTC 3213 : L ; TERM, C1
A477:	J3 + E0 500, 1536		; 1 FRTC 3214 : L ; TERM, C1
A478:	J3 + E0 501, 1536		; 1 FRTC 3215 : L ; TERM, C1
A479:	J2 + J3 + E11 1400 480		; 1 TR 3238 : L, L ; LO ; TERM
A480:	J2 + J3 + E11 1400 481		; 1 TR 3239 : L, L ; LO ; TERM

A481: J2 + J3 + E11 ; 1 TR 3240 : L, L
1400 ; LO
482 ; TERM
A482: J2 + J3 + E11 ; 1 TR 3241 : L, L
1400 ; LO
483 ; TERM
A483: J3 + E4 ; 1 FR 3216 : L
612, 759 ; TERM, C1
A484: J3 + E4 ; 1 FR 3217 : L
613, 759 ; TERM, C1
A485: J3 + E4 ; 1 FR 3218 : L
614, 759 ; TERM, C1
A486: J3 + E4 ; 1 FR 3219 : L
615, 759 ; TERM, C1

A487: J3 + E11 ; 1 TR 3242 : L
484 ; TERM
A488: J3 + E11 ; 1 TR 3243 : L
485 ; TERM
A489: J3 + E11 ; 1 TR 3244 : L
468 ; TERM
A490: J3 + E11 ; 1 TR 3245 : L
469 ; TERM
A491: J3 + E0 ; 1 TRC 3226 : L
630, 2048 ; TERM, C1
A492: J3 + E0 ; 1 TRC 3227 : L
631, 2048 ; TERM, C1
A493: J3 + E0 ; 1 TRC 3228 : L
632, 2048 ; TERM, C1
A494: J3 + E0 ; 1 TRC 3229 : L
633, 2048 ; TERM, C1
A495: J3 + E11 ; 1 TR 3355 : L
448 ; TERM
A496: J3 + E11 ; 1 TR 3356 : L
449 ; TERM
A497: J3 + E11 ; 1 TR 3357 : L
450 ; TERM
C0 ~ G12 ; CR, GROUP 12
A498: J3 + E0 ; 2 FRTC 3212 : L
504, 1536 ; TERM, C1
A499: J3 + E0 ; 2 FRTC 3213 : L
505, 1536 ; TERM, C1
A500: J3 + E0 ; 2 FRTC 3214 : L
506, 1536 ; TERM, C1
A501: J3 + E0 ; 2 FRTC 3215 : L
507, 1536 ; TERM, C1
A502: J2 + J3 + E11 ; 2 TR 3238 : L, L
1400 ; LO
470 ; TERM
A503: J2 + J3 + E11 ; 2 TR 3239 : L, L
1400 ; LO
471 ; TERM

A504: J2 + J3 + E11 ; 2 TR 3240 : L, L
1400 ; LO
472 ; TERM
A505: J2 + J3 + E11 ; 2 TR 3241 : L, L
1400 ; LO
473 ; TERM
A506: J3 + E4 ; 2 FR 3216 : L

A507:	644, 759	; TERM, C1
	J3 + E4	; 2 FR 3217 : L
	645, 759	; TERM, C1
A508:	J3 + E4	; 2 FR 3218 : L
	646, 759	; TERM, C1
A509:	J3 + E4	; 2 FR 3219 : L
	647, 759	; TERM, C1
A510:	J3 + E11	; 2 TR 3242 : L
	474	; TERM
A511:	J3 + E11	; 2 TR 3243 : L
	475	; TERM
A512:	J3 + E11	; 2 TR 3244 : L
	476	; TERM
A513:	J3 + E11	; 2 TR 3245 : L
	477	; TERM
A514:	J3 + E0	; 2 TRC 3226 : L
	662, 2048	; TERM, C1
A515:	J3 + E0	; 2 TRC 3227 : L
	663, 2048	; TERM, C1
A516:	J3 + E0	; 2 TR 3228 : L
	664, 2048	; TERM, C1
A517:	J3 + E0	; 2 TRC 3229 : L
	665, 2048	; TERM, C1
A518:	J3 + E11	; 2 TR 3355 : L
	451	; TERM
A519:	J3 + E11	; 2 TR 3356 : L
	452	; TERM
A520:	J3 + E11	; 2 TR 3357 : L
	453	; TERM
	C0 ~ G 13	; CR, GROUP 13
A521:	J3 + E0	; 3 FRTC 3212 : L
	510, 1536	; TERM, C1
A522:	J3 + E0	; 3 FRTC 3213 : L
	511, 1536	; TERM, C1
A523:	J3 + E0	; 3 FRTC 3214 : L
	512, 1536	; TERM, C1
A524:	J3 + E0	; 3 FRTC 3215 : L
	513, 1536	; TERM, C1
A525:	J2 + J3 + E11	; 3 TR 3238 : L, L
	1400	; LO
	488	; TERM
A526:	J2 + J3 + E11	; 3 TR 3239 : L, L
	1400	; LO
	489	; TERM
A527:	J2 + J3 + E11	; 3 TR 3240 : L, L
	1400	; LO
	490	; TERM
A528:	J2 + J3 + E11	; 3 TR 3241 : L, L
	1400	; LO
	491	; TERM
A529:	J3 + E4	; 3 FR 3216 : L
	676, 759	; TERM, C1
A530:	J3 + E4	; 3 FR 3217 : L
	677, 759	; TERM, C1
A531:	J3 + E4	; 3 FR 3218 : L
	678, 759	; TERM, C1
A532:	J3 + E4	; 3 FR 3219 : L
	679, 759	; TERM, C1
A533:	J3 + E11	; 3 TR 3242 : L

```

492 ; TERM
A534: J3 + E11 ; 3 TR 3243 : L
493 ; TERM
A535: J3 + E11 ; 3 TR 3244 : L
494 ; TERM
A536: J3 + E11 ; 3 TR 3245 : L
495 ; TERM
A537: J3 + E0 ; 3 TRC 3226 : L
694, 2048 ; TERM, C1

A538: J3 + E0 ; 3 TRC 3227 : L
695, 2048 ; TERM, C1
A539: J3 + E0 ; 3 TRC 3228 : L
696, 2048 ; TERM, C1
A540: J3 + E0 ; 3 TRC 3229 : L
697, 2048 ; TERM, C1
A541: J3 + E11 ; 3 TR 3355 : L
454 ; TERM
A542: J3 + E11 ; 3 TR 3356 : L
455 ; TERM
A543: J3 + E11 ; 3 TR 3357 : L
456 ; TERM
G0 ; GROUP ALWAYS ACTIVE
A544: J1 + J2 + E0 ; CHECK TEMP: H, L
7, -7 ; HI, LO
50, 1024 ; TERM, C1
A545: J1 + J2 + E0 ; CHECK 0-20 MV: H, L
505, 495 ; HI, LO
51, 1024 ; TERM, C1
A546: J1 + J2 + E0 ; CHECK 0-50 MV: H, L
505, 495 ; HI, LO
198, 1024 ; TERM, C1
A547: J1 + J2 + E0 ; CHECK 0-1 V: H, L
505, 495 ; HI, LO
229, 1024 ; TERM, C1
A548: J1 + J2 + E0 ; CHECK 0-10 V: H, L
505, 495 ; HI, LO
319, 1024 ; TERM, C1
A549: J1 + J2 + E0 ; CHECK TEMP: H, L
7, -7 ; HI, LO
399, 1024 ; TERM, C1
C2 ; END PART 2
```

H ; ALARM SCAN TABLE ITEM NO,SCAN NO

```

C9: 24 < 4 + 0 ; INITIAL ENTRY
      G1 ; GROUP 1
      1 < 4 + 0 ; PRSC 4813 ,561
      G2 ; GROUP 2
      2 < 4 + 0 ; 1LAL 1303 ,562
      3 < 4 + 0 ; 1LAH 1304 ,563
      4 < 4 + 0 ; 1PISCL 1316 ,564
      5 < 4 + 0 ; 1AISCH 1505 ,565
      6 < 4 + 0 ; 1LSCL 1509 ,566
      7 < 4 + 0 ; 1LSCH 1517 ,567
      8 < 4 + 0 ; 1LSCH 1605 ,568
      9 < 4 + 0 ; 1LICAH 1607 ,569
      G3 ; GROUP 3
      10 < 4 + 0 ; 2LAL 1303 ,570
      11 < 4 + 0 ; 2LAH 1304 ,571
      12 < 4 + 0 ; 2PISCL 1316 ,572
      13 < 4 + 0 ; 2AISCH 1505 ,573
      14 < 4 + 0 ; 2LSCL 1509 ,574
      15 < 4 + 0 ; 2LSCH 1517 ,575
      16 < 4 + 0 ; 2LSCH 1605 ,576
      17 < 4 + 0 ; 2LICAH 1607 ,577
      G4 ; GROUP 4
      18 < 4 + 0 ; 3LAL 1303 ,578
      19 < 4 + 0 ; 3LAH 1304 ,579
      20 < 4 + 0 ; 3PJSCCL 1316 ,580
      21 < 4 + 0 ; 3AISCH 1505 ,581
      22 < 4 + 0 ; 3LSCL 1509 ,582
      23 < 4 + 0 ; 3LSCH 1517 ,583
      1 < 4 + 2 ; 3LSCH 1605 ,584
      2 < 4 + 2 ; 3LICAH 1607 ,585
      G11 ; GROUP 11
      8 < 4 + 2 ; 1LPAS 3221 ,586
      9 < 4 + 2 ; 1LPAS 3222 ,587
      10 < 4 + 2 ; 1LPAS 3223 ,588
      11 < 4 + 2 ; 1LPAS 3224 ,589
      12 < 4 + 2 ; 1HLAS 3108 ,590
      13 < 4 + 2 ; 1HLAS 3109 ,591
      14 < 4 + 2 ; 1LLA 3112 ,592
      15 < 4 + 2 ; 1HLA 3112 ,593
      16 < 4 + 2 ; 1LTAS 3230 ,594
      17 < 4 + 2 ; 1LTAS 3231 ,595
      18 < 4 + 2 ; 1LTAS 3232 ,596
      19 < 4 + 2 ; 1LTAS 3233 ,597
      G12 ; GROUP 12
      1 < 4 + 4 ; 2LPAS 3221 ,598
      2 < 4 + 4 ; 2LPAS 3222 ,599
      3 < 4 + 4 ; 2LPAS 3223 ,600
      4 < 4 + 4 ; 2LPAS 3224 ,601
      5 < 4 + 4 ; 2HLAS 3108 ,602
      6 < 4 + 4 ; 2HLAS 3109 ,603
      7 < 4 + 4 ; 2LLA 3112 ,604
      8 < 4 + 4 ; 2HLA 3112 ,605
      9 < 4 + 4 ; 2LTAS 3230 ,606
      10 < 4 + 4 ; 2LTAS 3231 ,607
      11 < 4 + 4 ; 2LTAS 3232 ,608
      12 < 4 + 4 ; 2LTAS 3233 ,609
      G13 ; GROUP 13
      16 < 4 + 4 ; 3LPAS 3221 ,610

```

17	<	4	+	4	;	3LPAS	3222	,611
18	<	4	+	4	;	3LPAS	3223	,612
19	<	4	+	4	;	3LPAS	3224	,613
20	<	4	+	4	;	3HLAS	3108	,614
21	<	4	+	4	;	3HLAS	3109	,615
22	<	4	+	4	;	3LLA	3112	,616
23	<	4	+	4	;	3HLA	3112	,617
1	<	4	+	6	;	3LTAS	3230	,618
2	<	4	+	6	;	3LTAS	3231	,619
3	<	4	+	6	;	3LTAS	3232	,620
4	<	4	+	6	;	3LTAS	3233	,621
C3					;	END ALARM SCAN		

; PULSE SCAN TABLE PART 1

H C10:

C0 v G1

; CR AND GROUP 1

11 < 4 + 4	; 110 M 2001 , 631 : M 2
16 < 4 + 0	; 110 M 2002 , 632 : M 3
8 < 4 + 2	; 110 M 2003 , 633 : M 4
1 < 4 + 0	; 141 M 2001 , 634 : M 5
17 < 4 + 0	; 142 M 2001 , 635 : M 6
9 < 4 + 2	; 143 M 2001 , 636 : M 7
2 < 4 + 0	; 141 M 2002 , 637 : M 17
18 < 4 + 0	; 142 M 2002 , 638 : M 18
10 < 4 + 2	; 143 M 2002 , 639 : M 19
3 < 4 + 0	; 151 M 2001 , 640 : M 20
19 < 4 + 0	; 152 M 2001 , 641 : M 22
11 < 4 + 2	; 153 M 2001 , 642 : M 24
4 < 4 + 0	; 151 M 2002 , 643 : M 21
20 < 4 + 0	; 152 M 2002 , 644 : M 23
12 < 4 + 2	; 153 M 2002 , 645 : M 25
5 < 4 + 0	; 161 M 2001 , 646 : M 8
21 < 4 + 0	; 162 M 2001 , 647 : M 10
13 < 4 + 2	; 163 M 2001 , 648 : M 12
6 < 4 + 0	; 161 M 2002 , 649 : M 9
22 < 4 + 0	; 162 M 2002 , 650 : M 11
14 < 4 + 2	; 163 M 2002 , 651 : M 13
7 < 4 + 0	; 531 M 0301 , 652 : M 14
23 < 4 + 0	; 531 M 0302 , 653 : M 15
15 < 4 + 2	; 531 M 0303 , 654 : M 16
1 < 4 + 4	; 531 M 0304 , 655 : M 26
2 < 4 + 4	; 531 M 0305 , 656 : M 27
3 < 4 + 4	; 531 M 0306 , 657 : M 28

C0

; CR

14 < 4 + 0	; 410 M 2001 , 658 : M 47
4 < 4 + 2	; 410 M 2002 , 659 : M 48
20 < 4 + 2	; 410 M 2003 , 660 : M 49
21 < 4 + 2	; 410 M 2004 , 661 : M 50
8 < 4 + 4	; 410 M 2005 , 662 : M 51
15 < 4 + 0	; 420 M 2001 , 663 : M 52
5 < 4 + 2	; 420 M 2002 , 664 : M 53
6 < 4 + 2	; 420 M 2003 , 665 : M 54
22 < 4 + 2	; 420 M 2004 , 666 : M 55
7 < 4 + 2	; 420 M 2005 , 667 : M 56
23 < 4 + 2	; 420 M 2006 , 668 : M 57
9 < 4 + 4	; 420 M 2007 , 669 : M 58
10 < 4 + 4	; 100 M 2001 , 670 : M 59

C4

; PULSE SCAN TABLE PART 2

C11:

C0 v G5

; CR AND GROUP 5

8 < 4 + 0	; 211 M 2001 , 671 : M 34
12 < 4 + 4	; 212 M 2001 , 672 : M 35
16 < 4 + 2	; 213 M 2001 , 673 : M 36
4 < 4 + 4	; 214 M 2001 , 674 : M 37
9 < 4 + 0	; 211 M 2002 , 675 : M 39
17 < 4 + 2	; 213 M 2002 , 676 : M 40
10 < 4 + 0	; 532 M 0301 , 677 : M 29
1 < 4 + 2	; 532 M 0302 , 678 : M 30
18 < 4 + 2	; 532 M 0303 , 679 : M 31
5 < 4 + 4	; 532 M 0304 , 680 : M 32
6 < 4 + 4	; 532 M 0305 , 681 : M 33
11 < 4 + 0	; 532 M 0306 , 682 : M 38

G10	; GROUP 10
12 < 4 + 0	; 533 M 0301 , 683 : M 41
2 < 4 + 2	; 533 M 0302 , 684 : M 42
19 < 4 + 2	; 533 M 0303 , 685 : M 43
3 < 4 + 2	; 533 M 0304 , 686 : M 44
13 < 4 + 0	; 533 M 0305 , 687 : M 45
7 < 4 + 4	; 533 M 0306 , 688 : M 46
16 < 4 + 4	; WI 3501 , 689 : M 60
17 < 4 + 4	; WI 3502 , 690 : M 61
18 < 4 + 4	; WI 3503 , 691 : M 62
19 < 4 + 4	; WI 3504 , 692 : M 63
20 < 4 + 4	; WI 3505 , 693 : M 64
21 < 4 + 4	; WI 3506 , 694 : M 65
22 < 4 + 4	; WI 3507 , 695 : M 66
23 < 4 + 4	; WI 3508 , 696 : M 67
C5	; END PULSE SCAN TABLE

; PUNCH OFF CODE FOR THE MACRO ASSEMBLER:

```

S E0, K=0 ; TAIL SEGMENT 11:
W E0: AL W0 A282 ; INITIALIZE(
      HS W0 I0 ; H1,
      AL W0 A283 ;
      HS W0 I0+1 ; H2,
      AL W0 A284 ;
      HS W0 I0+2 ; H3,
      AL W1 C7 ;
      AL. W0 X1+E0. ;
      RS W0 I1 ; B30,
      AL W0 C8 ;
      RS W0 I2 ; B31,
      AL W1 C9 ;
      AL. W0 X1+E0. ;
      RS W0 I3 ; B32,
      AL W1 C10 ;
      AL. W0 X1+E0. ;
      RS W0 I4 ; B33,
      AL W1 C11 ;
      AL. W0 X1+E0. ;
      RS W0 I5 ; B15);
      AL. W0 E0. ;
      LD W2 48 ;
      JL W3 (I7) ; CALL LOADER;
      JL. E0. ;

```

```

E
W C6: AL. W0 C6.
      AL W1 0
      AL W2 I6>22^1 ;
      JL W3 (I7) ; CALL LOADER;
      JL. C6.

```

```

L
; IDLIST S11
N

```

```

T
E
E
L
;END S11
N

```

```

E ; END OF PULAWY PROGRAM SYSTEM

```

S D2, E5, G3, H40, I10, J20, K=0

```
; THE MACRO ASSEMBLER PROGRAM IS PART
; OF THE PULAWY PROGRAM SYSTEM, AND SERVES
; FOR EASY ASSEMBLY OF THE 6-TH SEGMENT OF IT.
; THE PROGRAM ACCEPTS FOR INPUT AN
; UPDATED VERSION OF SEGMENT 11 (THE SCAN TABLE)
; AND A SEGMENT 6 IN WHICH CORRECTIONS
; WERE MADE ONLY IN THE COMMENT FIELD.
; THE PUNCHED OUTPUT WILL BE AN OPERATIONAL
; SEGMENT 6.
; IF THE E2 AND/OR E3 PARAMETERS OF
; THE SYSTEM WILL BE CHANGED, THIS PROGRAM
; MUST BE RECOMPILED AFTER CORRECTING THE
; FOLLOWING DEFINITIONS:
```

```
E2 = 270 ; NUMBER OF BALANCE RESULTS
E3 = 14 ; - - PLANT DATA
E4 = 125 ; BASE FOR W
E5 = 109 ; - - E
```

W G0: 1<20 ; BIT PAPER OUT

G1: 10 ;

G2: 15 ; MASK OPCODE

I0: 0 ; INCHAR

I5: 0 ; OUTCHAR

I9: 0 ; RETURN

H I1: 0 ; CASE

J0: <

SCAN TABLE, PLEASE<10>>

J1: <

SOURCE PROGRAM, PLEASE <10>>

J3: <<63><0><0><0>;<74>>

J9: <<0><0><0><0><0><0><0><0><0><10>>

J10:<DDDDDDDD<10>>

J11:<<0>,<10>>

J4: <TR>,<AR>,<TS>,<ST>,<SW>,<MU>,<DI>,<TF>,<AF>,<TC>,<EX>,<TM>,<AM>,<SR>

J5=K-J4

J7: 0, 1, 130, 3, 4, 5, 6, 135, 136, 137, 10, 267, 268, 13

J8: 0,0,0,0,0,0,0,0

J6: <

SYNTAX <10>>

J2: <<58><31><63><10>>

J12:<

END<10>>

W H1: SE W0 58 ;

AM 128 ;

AL W0 0 ;

HS. W0 I1. ;

H H0: A50 , A31 ; PROCEDURE INCHAR;

I0. , A22+A0 ; BEGIN INTEGER CHAR;

A50 , A31 ;

I0. , A20+A0 ; L: CHAR := IO(READER);

W RL. W0 I0. ;

SZ. W0 (G0.) ; IF INTERVENTION REQUIRED

JL. H31. ; THEN GOTO PAPER OUT;

SH W0 64 ; IF OTHER ERROR THEN

JL. H2. ; BEGIN

AL W1 A0 ; DEVICE ERROR (READER)

H4: AL W0 0 ; GOTO RETURN

JL W3 (B21) ;

JL. (I9.) ; END;

H H2: SO W0 , I4: 0 ; BOOLEAN COPY;

W JL. H13. ;

```

A50      , A31      ; IO(PUNCH, CHAR);
I5.      , A1+A20   ; IF CHAR = 58 ~ CHAR = 60
W        RL. W0     I5.      ; THEN BEGIN
        SH W0       64      ; CASE := IF CHAR = 58
        JL.        H14.    ; THEN 0 ELSE 128;
H16: AL W1         A1      ; GOTO L END;
        JL.        H4.     ; INCHAR:= CHAR +
H14: RL W0         I0.     ; CASE
H13: SE W0         58     ; END INCHAR;
        SN W0       60     ;
        JL.        H1.     ;
        SN W0       0      ;
        JL.        H0.     ;
        BA. W0     I1.     ;
        JL          X3     ;

H15: AL W3         0      ; PROCEDURE OUTTEXT;
H17: AC W3         X3+6    ;
        BZ W0       X2     ;
        LS W0       X3+18  ;
        LS W0       -18    ;
        SN W3       0      ;
        AL W2       X2+1   ;
        SN W0       10     ;
        JL          X1     ;
        SN W0       63     ;
        AL W0       64     ;
        A50      , A31     ;
        I5.      , A1+A23  ;
        A50      , A31     ;
        I5.      , A1+A20  ;
W        RL. W0     I5.     ;
        SH W0       65     ;
        JL.        H17.    ;
        JL.        H16.    ;

H32: AL W1         -1     ; BEGIN:
        HS. W1      I4.     ; I:=J:=-1;
        HS. W1      I2.     ; COPY:=FALSE;
        HS. W1      I3.     ;
        AL. W0      0      ; X:
        RS. W0      I9.     ; RETURN:=X;
        AL. W0      J0.     ; WRITETEXT(<<SCAN TABLE, PLEASE>>);
        JL W3      (B26)   ;
        JL W3      (B27)   ; TYPECHAR;
        JL. W3      H0.     ; L: K:= INCHAR;
        SE W0      31     ; IF K = 31 THEN
        JL.        -4     ; GOTO L;
H6: AL W1         -1     ;
        AL W2       -1     ; N: AC:= ASC:= FALSE;
H7: JL W3         H0.     ; M: K:= INCHAR;
        SN W0      31     ; IF K = 31 THEN
        JL.        H12.    ; GOTO U;
        SZ W0      64     ; IF K = <CR> THEN
        JL.        H6.     ; GOTO N;
        SN W0      128+5   ; IF K = <;> THEN
        AL W2      0      ; ASC:= TRUE;
        SN W0      128+59  ; IF K = <:> THEN
        AL W1      X2     ; AC:= ASC;
        SE W1      0      ; IF -, AC THEN
        JL.        H7.     ; GOTO M;
        SN W0      128+57  ; IF K = <I> THEN
        JL.        H8.     ; GOTO INTEGR;
        SE W0      128+36  ; IF K = <M> THEN
        JL.        H7.     ; GOTO M;
        ; INTEGR:
H8: AL W2         (0)    ; NUMB:= 0;
        AL W1      0      ; SAVE:= K;
H9: JL W3         H0.     ; DIGIT: K:= INCHAR
        SN W0      16     ; IF K = 16 THEN

```

```

      JL.      H10.      ; GOTO END NUMB;
      AL W3      (0)      ; NUMB:= 10 * NUMB + K;
      WM. W1      G1.      ;
      WA W1      6        ;
      JL.      H9.      ; GOTO DIGIT;
H10: SE W2      128+36    ; END NUMB:
      JL.      H11.      ; IF SAVE = <M THEN
      AL W2      , I2: 0   ;
      AL W2      X2+1     ; M[NUMB]:= J:= J + 1
      HS. W2      I2.      ;
      HS. W2      X1+D2.   ;
      JL.      H6.      ; ELSE
H11: AL W2      , I3: 0   ; F[NUMB]:= I:= I + 1;
      AL W2      X2+1     ; GOTO N;
      HS. W2      I3.      ;
      HS. W2      X1+D1.   ;
      JL.      H6.      ;
H12: AL W0      0        ; U:
      RS. W0      I9.      ; RETURN:= U;
      AL W0      J1.      ;
      JL W3      (B26)     ; WRITETEXT(<SOURCE PROGRAM>);
      JL W3      (B27)     ; TYPECHAR;
      AL W0      0        ; COPY:= TRUE;
      HS. W0      I4.      ;
      JL. W3      H0.      ; WHILE INCHAR = 31 DO;
      SE W0      31       ;
      JL.      -4        ;
H28: AL W0      -1       ; NEXT LINE:
      HS. W0      I4.      ; COPY:= FALSE;
H29: JL W3      H0.      ; NEXTCHAR: U:= INCHAR;
      SN W0      31       ; IF K = 31 THEN
      JL.      H30.      ; GOTO COPYTAIL;
      SE W0      128+5    ; IF K = <;> THEN
      JL.      H29.      ; GOTO NEXT CHAR;
      AL W2      J3.      ; OUTTEXT(<,,,,;>);
      JL. W1      H15.     ;
      AL W0      0        ;
      HS. W0      I4.      ; COPY:= TRUE;
      HS. W0      I6.      ; I:= 0;
H19: JL W3      H0.      ; ASSEMBLY: K:= INCHAR;
      SE W0      16       ; IF K = <0> ^ K < 10 THEN
      SH W0      9        ;
      JL.      H18.      ; GOTO COMMENT;
      BS. W0      I1.      ; K:= K - CASE;
      LS W0      6        ;
      HS. W0      I2.      ; TEMP:= K SHIFT 6;
      JL. W3      H0.      ; K:= INCHAR;
      BS. W0      I1.      ; K:= K - CASE + TEMP;
      BA. W0      I2.      ;
      AL W3      0        ; FOR N:= 0 STEP 1 UNTIL
H21: BL W1      X3+J4.    ; TOP DO
      SN W0      X1       ; IF INSTR[N] = K THEN
      JL.      H20.      ; GOTO MACRO FOUND;
      AL W3      X3+1     ;
      SE W3      J5       ;
      JL.      H21.      ;
      AL W0      J6.      ; WRITETEXT (<SYNTAX>);
      JL W3      (B26)     ; GOTO ASSEMBLY;
      JL.      H19.      ;
H20: BZ. W3      X3+J7.   ; MACRO FOUND:
      HS. W3      I2.      ; PART 1:= INSTR VALUE[N];
      AL W2      0        ; ADDRESS:= INCR:= 0;
      HS. W2      I7.      ;
H23: JL W3      H0.      ; ADDRESS: K:= INCHAR;
      SN W0      16       ; IF K = 16 THEN
      AL W0      0        ; K:= 0;
      SL W0      10       ; IF K > 9 THEN
      JL.      H22.      ; GOTO END ADDRESS;
      WM. W2      G1.      ; ADDRESS:=
      WA W2      0        ; 10 * ADDRESS + K;

```


RS	W0	A47+16	;
AL.	W0	E0.	;
AL	W1	0	;
AL	W2	1	;
JL	W3	(B39)	;

E1: <MACR>	;	NAME
JL.	E0.	;

E
L
; IDLIST MACRO ASSEMBLER

N
T
E
L
; END MACRO ASSEMBLER
N
T